UiO : Department of Informatics
University of Oslo

Cognitive Entity Authentication with Petname Systems

Kent Are Varmedal, master thesis spring 2013

Cognitive Entity Authentication with Petname Systems

Kent Are Varmedal

Spring 2013

Abstract

In cybercrime, phishing attacks are amongst the most popular methods to acquire information about individuals. An attack to gather user names and passwords can be preformed by making a fake login website, similar to a legitimate web service, and distribute its hyperlink to unsuspecting users. Our goal in this thesis was to design and test a solution that would reduce the number of victims of phishing attacks.

Service authentication on the Internet mainly consists of identity corroboration, e.g. certificates. However even if the identity is correct, it is no guarantee that the identified service is one the user wants to access.

Cognitive entity authentication is a process where the security policy in ordinary authentication processes is replaced with decisions preformed by an entity with cognitive abilities, e.g. a human. It is time consuming and difficult for a human user to do cognitive authenticate a service on the internet, i.e. verifying that the service is the right one and not some fraudulent site. We have designed and implemented a Petname System which gives the user the opportunity to add personal Petnames for the services he or she uses. These Petnames can afterwards be used by the user to verify if the accessed service is the same as the last time. The Petname System help will limit the rate of successful phishing attacks.

Our Petname System is an external device, which lets the user take the Petnames with them and use the same system on different computers and platforms. Our Petname System did not cover all the properties set for such systems in earlier research; this was mainly because the system is external and not a part of the web browser.

The user test of our Petname System was positive. All the subjects that did not notice our phishing sites were stopped by our Petname System.The subjects understood the problem of phishing, and saw the Petname System as an extra precaution when surfing the Internet. The subjects were of the opinion that the system should be on a device which they are already using, e.g. a mobile phone. They would not use a separate device only for the Petname System; it had to at least provide a number of different security services.

Acknowledgement

I would like to thank to my supervisor Professor Audun Jøsang for giving me such an interesting assignment, which also were a part of an international research project. It is not usual and I greatly appreciate the experience of collaboration across borders and cultures, including the opportunity to participate in two published papers and be a lead author for another published paper. This could not have been done without his support.

I would also like to give my thanks to Henning Klevjer for all the good input and discussions throughout this work. As we were working on the same device a number of our challenges were similar.

To Åsmund Aasen Devold for all the support and patience he has shown while I have been working on this project.

Maria Anker Middelthon for all the good discussions and input during this work, especially regarding the interviews.

And the rest of my friends and proofreaders on the lab, Ali Ahmad, Magnus Evensberget, and Kristoffer Jensen, for making this time of hard work fun and enjoyable. Without their input this thesis would most likely look like a manuscript from the advent calendar TV series "The Julekalender".

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Authentication is often considered as something a user does to a service. It can also be authentication of the service itself or of the origin of a message. In this thesis we will focus on how a user can authenticate a service, e.g. a website.

There are multiple ways for an attacker to guide a user to a fake website, as a method to get the user to divulge some personal information. This type of attacks is often called phishing attacks. Users are often not aware of this threat and will not react to subtle differences between the fake and the genuine website. A user can use an application to help check for tell tale signs of a phishing attack, such as a Petname System which is studied in this thesis.

Petname Systems can be described simply as a way to allow users to assign petnames to a spesific server system. These server systems can easily be recognised by the petnames, there of the name.

Ferdous, Jøsang, Singh and Borgaonkar [22] propose to implement a Petname System on a secure mobile device. As a solution to make such systems readily available for the user at any time and any place, and not restrict them to one specific browser installation.

1.1 Assignment

This master thesis describes the theory behind cognitive entity authentication based on Petname Systems. We will implement an external Petname System and evaluate its usability by doing user tests and interviews.

1.2 Methodologies

Computer science is a relatively new field of research compared to other disciplines. Nevertheless it includes elements from nearly every other field of study. We will now highlight some of the most common methodologies used in computer science.

1.2.1 Theoretical

Theoretical computer science relates strongly to mathematics and logic. In this branch of computer sciences hypothesises are formally proven or disproved by the use of mathematical, logical and combinatorial theories. In this category we find the constructing and analysis of algorithms.

1.2.2 Simulation

Computers have a central place in simulations of any kind. In everything from physics and logistics to weather predictions and population growth. The models of such simulations are based on theory with numerous variables depending on each other. The complexity makes it practically impossible to do by hand. One big benefit to computer simulations is that one can introduce a fault in the model to see how it affects the result, without the consequences this might give in a real-world scenario.

1.2.3 Development

In computer science the development of systems and prototypes is the most important method to test and evaluate solutions. It is this that brings the theoretical science to life. There is also a number of different models of development. One example is the Waterfall model that is a sequential process going through every step of the development once to the end of the project. Another is the Iterative method, which develops a part of the system over a limited amount of time, evaluates the results and starts a new iteration.

1.2.4 User interaction

The human side of computer science is focused on user interaction and experience and the scientific understanding of these. Often research into this area is used to make a system more user-friendly, which is to develop a more intuitive and easy system for the user. It also includes how persons with disabilities interact with computers and software.

1.2.5 In this project

Our project is a combination of development and user tests. The first part is to develop a mobile Petname System. Here we will use the extreme prototyping model of software development. This model is often used in website development and consists of three phases.

1. Make a static prototype.
2. Make the prototype fully functional, with simulated service layer.
3. Make the service layer.

The process would probably differ in one way, some internal workings on the service layer would be made before the functionality in the user interface is implemented. The last phase is mainly to get it to communicate with the web browser.

After the development is done we would continue with user tests, interviews and the analysis of the results. It is important to get input from user experience and if there is any increased security gained from the Petname System.

1.3 Motivation

The problem of phishing is growing. Most of the organisations subjected to such attacks try to stop the attacker from getting into their systems. There is little focus on how the users can protect their own information from getting into the hands of an attacker.

A Petname System is one way to help the user identify their known services, by giving the user the opportunity to add a personal Petname to the service. Some Petname Systems already exist but are not widely used. The reasons might be that they are not widely known or the systems is too difficult to use. Existing solutions are tied to a specific browser installation. Which makes it hard to transfer one set of Petnames from one browser to another on the same or other computers.

This is the particular problem for which we would like to propose a solution. It is also important to test the users understanding of the problem and if they would use such a system if it was available to them. It might also give an indication of how willing users are to use new systems to ensure the security of their personal computing environment.

1.4 Research Questions

In this thesis we will answer four questions. The first is to find which obstacles stands in the way for cognitive entity authentication on the Internet today. It leads nicely into the next question, how a Petname System can aid the user in the process of cognitive server authentication.

There is no existing external Petname System available, so it has to be designed and implemented. Lastly we need to find out if such a Petname System would be used by users.

- What stands in the way of Cognitive entity authentication?
- How can a Petname System help with Cognitive entity authentication?
- How can an external Petname System be designed?
- Would users use an external Petname System?

1.5 Related Work

1.5.1 Browser extensions/plugins

In [22] the authors evaluate two different extensions to a web browser, *Petname Tool* and *TrustBar*.

None of these solutions are updated and both can be considered defunct. *Petname Tool* was last updated on the 30th of June 2009 [16], while it is hard to say when *TrustBar* was last updated. If [35] is anything to go by, the last update was late january 2006.

It is unclear why they stopped developing these extensions, as there are many factors to consider. For instance number of users, the demand and understanding of such systems and resources available to the developer.

A careful search for other in-browser Petname System has returned false for all the major web browsers[1]. There is a possibility that Petname or similar systems are included into other browser extensions without it being mentioned in the descriptions or anywhere else.

1.5.2 External secure devices

A number of secure devices has been described in the literature. Some devices are designed to replace password, others can incorporate several complete security systems. The focus on such devices by different researches around the world shows that there is a need of an extra device to get higher security. We will now describe some of the relevant devices.

OffPAD

The OffPAD is an Offline version of the PAD (Personal authentication device) proposed by Jøsang and Pope [43] as a solution for user centric identity management. In contrast to other models of identity management, e.g. federated and centralised user identity model, different user credentials are stored in the PAD which is in the user's personal domain. The user authenticates to the PAD and the PAD authenticates the user to the service.

The idea of the PAD has evolved into a multi purpose security device, able to host a number of different applications. It should be offline in the sense that it is only connected when the user wants to use it [42].

Nebuchadnezzar

Based on a concept similar to the OffPAD, Laurie and Singer describe in their position paper [51] the Nebuchadnezzar. It is a secure device used for authentication of users and messages. The focus of their paper is to define the requirements for the operating system of such devices. They also give two examples of usage scenarios: user authentication and transaction authorisation. The latter lets the user see a requested transaction and deciding if he or she should authorise it for further processing.

[1] Internet Explorer, Google Chrome, Mozilla Firefox, Opera and Safari

Pico

The Pico device proposed by Frank Stajano, is designed to replace passwords everywhere [71]. The device requires that servers, applications and computers support the protocol and the device communication. These requirements alone makes it hard to get this solution into public use.

Stajano introduced a new way of authentication called *Picosiblings*. The idea is that the Pico only opens in a "friendly" environment, in the proximity of its *Picosiblings*. These siblings have a small transmitter and can be put into everyday items like glasses, belt buckles, watches and so on. He does not address the privacy issues introduced with the possibility discover persons having *picosiblings*.

1.6 Report Outline

The overall structure of this document is first theory, then discussion and lastly conclusion. The theory part consists of this chapter and Chapter 2 Background, where we describe ideas, technologies and research that is relevant for this project. It will start with an introduction to authentication and cognitive entity authentication. We will look on the technical aspects of authentication and the internet.

Discussion is done in three chapters. The first of these is Chapter 3 General Discussion. Where the main topics from the previous chapter is discussed. The design and implementation of our external Petname System is described in Chapter 4 Technical Description. It also includes which considerations and limitations we had to address in the development.

As a practical evaluation of the system, we have performed user tests and interviews. The different aspects of the planning and execution of the evaluation is described in Chapter 5 User Test. In addition the result of the tests and interviews is presented at the end of the chapter.

We summarise our work and results in Chapter 6 Conclusion. Throughout this work we have come across several ideas that is either a continuation of this work or is relevant for future development. All these are put together in Chapter 7 Future work.

1.7 The Lucidman Project

Our thesis is a part of the Lucidman (Local User Centric ID Management) project [53]. The project is done in cooperation between Norwegian and French companies and higher education institutions.

In addition to the University of Oslo the following companies and higher education institutions was a part of this project: GREYC, TazTag, CEV, Tellu, and Vallvi

The aim for this project is to focus on the identity management on the client side in a simple and usable method. There is a big challenge in how to ensure good security and usability for the user, as these aspects often are considered contradictory.

The project started in the spring of 2011 with a time span of two years. During this time there was monthly telephone meetings and semi-annual face-to-face meetings, alternating between Oslo and Normandy. The project had a budget of 530.000 € and resulted in a number of papers, two patent applications, two open workshops and multiple master theses.

1.7.1 GREYC

GREYC is a research lab that is a part of ENSICAEN, which is one of the national universities of engineering in France. Degrees awarded from institutions like ENSICAEN are internationally regarded as equal to a Master's in science.

It is the E-Payment and Biometrics Research Unit at GREYC that participate in the LUCIDMAN project. The unit is directed by Professor Christophe Rosenberger and mainly works within computer security. They have a focus on biometrics and trust. In the LUCIDMAN project they contributed with biometric use cases and development of biometric applications.

1.7.2 TazTag

TazTag is a company that produces secure electronic devices. Some of the devices have biometric capabilities. They are developing pads and phones with higher security by introducing secure hardware elements available for the developers of applications. They contributed in this project with devices, like the TazCard and TazPad.

1.7.3 CEV

CEV makes card solutions for discrete payment applications. includes prepaid vouchers, travel tickets and loyalty cards. Their most known product is the shopping card Cartaplus, which is widely spread around the world. CEV is a part of Chèque Déjeuner Group in France, which is the biggest provider of prepaid meal vouchers in France. In this project they contributed with business loyalty and e-shopping use cases, as well as being the French project manager.

1.7.4 Vallvi AS

Vallvi is a Norwegian company doing consultant project management and business development within the field of information and communication technology. Vallvi provides the Norwegian project manager of the LUCIDMAN project.

1.7.5 Tellu AS

Tellu is a small Norwegian IT company that both develops their own products and provide consulting services. They specialise in mobile

applications and tracking, which is used in their SmartTracker product. It is used by physical security companies to tell if they have checked physical access points (e.g. doors and gates). In LUCIDMAN they focused on use cases and development.

Chapter 2

Background

In this chapter we will go through ideas, research, articles and other relevant material for our thesis. As authentication is an important part of this thesis as well as every day life, we will start with theory about authentication and how server authentication is done today.

Before we go into the topic of phishing in Section 2.4 on page 25, we will take a short look on the development of the World Wide Web, as it is highly relevant.

One of the techniques that can help against the threat of phishing attacks is Petname Systems. In Section 2.5 on page 29 we will describe the model these systems builds upon and different work done on Petname Systems. We will also go into different secure devices that can be used by such a system in Section 2.6 on page 34.

2.1 Authentication

When discussing authentication, the topic is often how to authenticate a client to a service or other resources of some kind. For instance the act of providing user credentials for on-line authentication is now becoming second nature for any internet user. The field of authentication is more than just user authentication, in Figure 2.1 on the next page we can see the taxonomy of authentication.

In the X.800 standard [14] there are two types of authentication services, *Peer-entity authentication* and *Data origin authentication*.

Data origin authentication
The corroboration that the source of data received is as claimed.

Peer-entity authentication
The corroboration that a peer entity in an association is the one claimed.

The *data origin authentication* service provides the corroboration for the source of the data. It must not be confused with data integrity as the

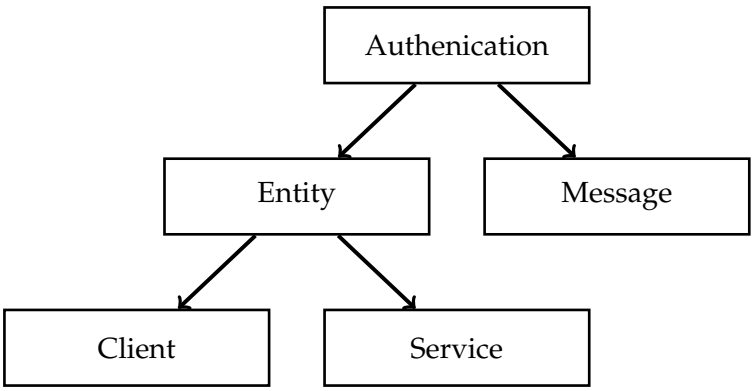

Figure 2.1: Taxonomy of authentication

data can be modified as well as duplicated while the source identity is corroborated.

Peer-entity authentication is a service that provides identity confirmation of communicating entities. It can only be provided under the establishment of a connection and while transferring data. Depending on the authentication scheme this can be just one way or a mutual peer entity authentication.

X.800 is related to the Open Systems Interconnection (OSI) Basic reference model [13]. The authentication provides corroboration of the identity to the layer above the layer where the service itself is provided. For instance a service on the network layer (layer number 3), can provide corroboration of the identity to the transport layer (number 4). It should be noted that this is an overarching description of how such a system should work, not a guide for implementation.

The X.800 standard is describing what we call syntactic authentication, where it only check if the peer entity or the data origin are what they say that they should be. It does not concern itself with the nature of the entity nor the security policy. So an entity can authenticate itself as the Mafia and it would not be anything different from any other authenticated entity.

When peer entities are discussed, it is the server and the client that are under consideration. Often forgetting that normally servers and clients are only tools for organisations and users. Changing the peer entities to be a user and an organisation adds several layers of complexity to the discussion.

Figure 2.2 on the facing page shows different types of entity authentication with the four entities *Service Provider Organisation*, *Server system*, *Client system* and *Human User*. They can authenticate each other between the *User Side* and *Server Side*. The different types of entity authentication is described below.

2.1.1 User authentication

There are a number of ways for a server to authenticate a user. The simplest way is to use a user name and password. This has been the way to do user authentication from the early years of the Internet. Still web forums and

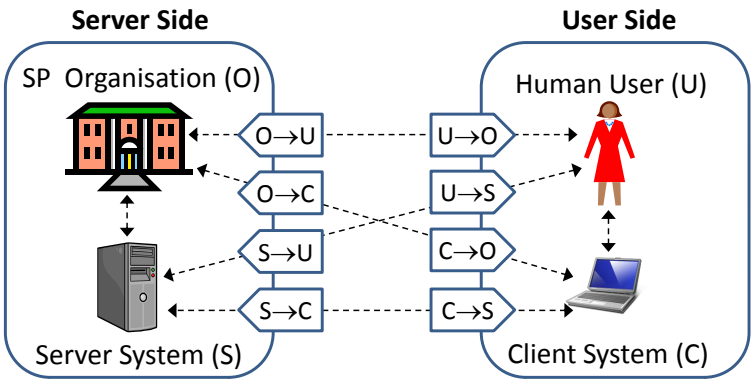

Figure 2.2: General entity authentication types [78].

User side authentication

$U \rightarrow O$: The User is authenticated by the Organisation.
$U \rightarrow S$: The User is authenticated by the Server.
$C \rightarrow O$: The Client is authenticated by the Organisation.
$C \rightarrow S$: The Client is authenticated by the Server.

Server side authentication

$O \rightarrow U$: The Organisation is authenticated by the User.
$O \rightarrow C$: The Organisation is authenticated by the Client.
$S \rightarrow U$: The Server is authenticated by the User.
$S \rightarrow C$: The Server is authenticated by the Client.

other none sensitive solutions often use only user name and password.

Services with more sensitive information often use user names and passwords in combination with other factors of authentication. The reason why this is done is to give a higher level of confidence in the authentication process. Two-factor authentication is when an authentication method requires two types of credentials. These credentials can be placed into different categories.

Something you know
Passwords, PIN, etc.

Something you are
Fingerprints, voice, key-stroke dynamics and other biometrics.

Something you have
Code lists, One-time pads, smart cards, applications on smart phones, other physical and virtual devices.

2.1.2 Cognitive entity authentication

In [41] we coined the term Cognitive entity authentication. It is used to describe authentication done by human users and organisations. A human is considered to be a cognitive entity because of its non-deterministic free will. An organisation is also regarded as a cognitive entity because it is governed by one or more humans. Clients and servers is considered as system entities, as they only do what they are programmed to do.

Since we did not give the term Cognitive entity authentication a clear definition in [41], we will define it now.

The National Institute of Standards and Technology (NIST) defines identification as the process whereby a network element recognises a valid user's identity. Authentication is defined as the process of verifying the claimed identity of a user [64]. We will call this type of authentication *Syntactic entity authentication*.

Syntactic entity authentication
The verification by the relying party that the identity of the entity in a communication session is as claimed.

When a system combines *Syntactic entity authentication* and verification of required characteristics of the entity, we get *Semantic entity authentication*. For instance verifying that the identity is on a white list or has the required reputation level.

Semantic entity authentication
The verification by the relying entity (A) that the identity of the entity (B) in a communication session is as claimed, and in addition the verification by entity A that entity B has semantic characteristics that are compatible with a formal security policy.

In Cognitive entity authentication the formal security policy from the Semantic entity authentication is replaced with cognitive reasoning capability. The cognitive entity authentication is normally performed by humans[1], who apply their cognitive reasoning capability to examine the entity and make a decision.

Cognitive entity authentication
The verification by the cognitive relying party (A) that the identity of the entity (B) in a communication session is as claimed, and in addition the examination by entity A of the true nature of entity B in order to decide if it is acceptable to connect to the authenticated entity.

System entities can easily authenticate each other with methods and technologies such as cryptographic certificates. It can be used on both sides to ensure mutual system entity authentication. The client can save a hash of the server's certificate, which can be used to verify that the certificate is the same as last time.

A problem arises when a cognitive entity should authenticate a system entity, usually a human user authenticating the server. This is not trivial to do at the moment, as there are plenty of ways to fool either the client or the user. To fool the system entity client, an attacker needs to introduce some malicious information in an insecure part of the client or the systems the client relies upon.

To fool a human is easier. Our brain has the capability to add missing information and change perceived information. Such effects are illustrated by the large number of optical illusions, where the brain perceives the reality differently then it actually is.

Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Some years ago the text in the previous paragraph circulated on the internet. It is a practical example that shows that the brain sees what it expects to see. Matt Davis, a researcher at the *Cognition and Brain Sciences Unit* in Cambridge UK, has written a web page about this effect [18]. He points out that the text is probably manipulated to be easy to read and gives a list over techniques to accomplish this.

A widely used technique is to change a character with another that is similar to the first. The classic example is *"Paypal"* and *"Paypa1"*. The success of such replacements is largely depending on the font used.

[1] It is also possible to construct artificial intelligence to do this in specific cases.

By using character replacement and order changes an attacker could be able to register a domain name. The attacker can guide unsuspecting users to the website using this domain name. This calls for a user friendly way to easily check the identity of a service, a way to do a cognitive server authentication.

2.2 Authentication of Servers

There is currently no easy way for a user to authenticate the servers he or she accesses. In a web browser it is possible to double check domain name, SSL certificate or other properties. However it is difficult, time consuming and can be forged. We should not dismiss the mental load the users would be experiencing if they had to check and remember key information for every service they interact with. It is important that the authentication process gives the user as little mental load as possible. If this load is too high people would not perform authentication, as humans normally choose the path of least resistance.

In the Indian framework for e-Authentication, called *e-Pramaan* [21], they have a section about website authentication. They suggest the following techniques to prevent phishing:

- User education and awareness
- Web site design to avoid phishing:
  - Watermark/Customised logo
  - Last login details, last transaction details, etc.
- Digital certificates for Web sites
- Programming solutions to prevent superimposition by face Web sites.

The framework is shorter than its draft [32], which gives better descriptions for the different techniques mentioned above. We will go more into what this draft describes in relation to Cognitive entity authentication in Section 2.2.4 on page 21.

2.2.1 The domain name system

In the beginning when the internet was small, the users could remember the IP-addresses for their services. This is no longer the case. To find our way around the internet we have to use the Domain Name System (DNS). The domain names in this system are just pointers to IP-addresses and work like a public distributed hierarchical one-way phone book.

Lookup of the IP-address to the related domain name, might be harder. It requires that a reverse DNS pointer is added. Such a pointer has to be added by an internet service provider that owns the IP-address. For instance such a pointer is added for the IP-address 129.240.8.200. In the

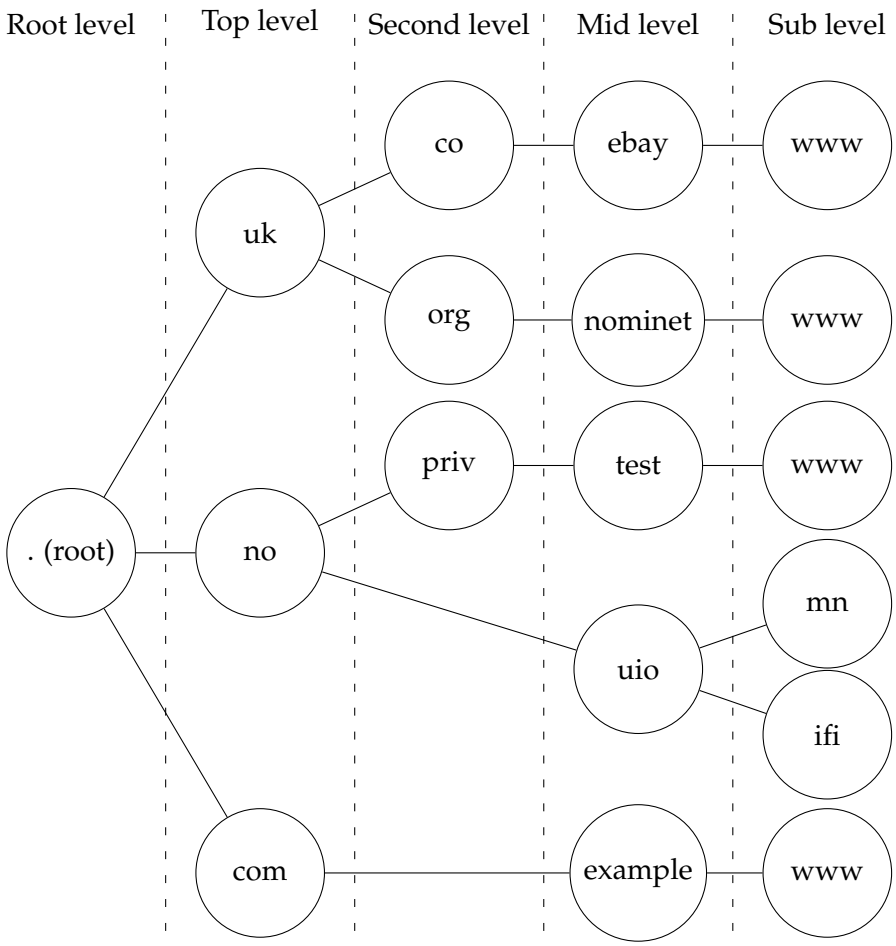

Figure 2.3: The hierarchical domain name system

DNS the domain name "200.8.240.129.in-addr.arpa"[2] returns a pointer to www.uio.no.

It is not mandatory to have a reverse pointer. Because one web server can host thousands of domain names the pointer is often impractical to maintain. However there are services on the Internet that provide such information even if the reverse record does not exist. The information is gathered in databases by saving which IP-address a domain name points to.

The hierarchy

The root domain is controlled by Internet Assigned Numbers Authority (IANA)[3]. They are the top of the hierarchy and have distributed the control of each top level domain name to countries, territories and private

[2]The IP-address is written in reverse in the domain name to be able to delegate ranges of IP-addresses to different name servers.

[3]http://www.iana.org/

agencies. For instance *.com* to the company Verisign[4], *.no* to Norway where UNINETT Norid AS[5] is the appointed registry by the government and *.as* to American Samoa handled by AS Domain Registry[6]. Some top level domains have a second level of domain names, with special use like *.co.uk* intended for commercial enterprises [61] and *.priv.no* is only for private individuals registered in the Norwegian National Register [6]. A part of the DNS structure is shown in Figure 2.3 on the preceding page.

There are also companies that sell sub domains without being connected to the official registries, e.g. *.co.no* where a company have bought the rights to the *co.no* domain name. They have made it possible for anyone to register sub domains for this domain name [12], giving foreign companies the opportunity to register a "Norwegian" domain name.

It is easy to register a domain name and they can be quite similar to already existing domain names. Different companies often have the same name in different domains, e.g. telenor.no and telenor.se. Such naming schemes make users more accustomed to see other domain names for the same service or services from the same entity.

DNS poisoning

It is possible to corrupt a domain name server cache, making subsequent queries to the DNS return wrong data. A corruption like this would make the client connect to a bogus and in most cases malicious server. To explain how this happens we first need to know how a DNS server works.

When a DNS-resolver queries a *Name Server* for a DNS-record, it sends a UDP-packet with a unique 16-bit identifier. The server response includes this identifier to match up the reply to one request. The identifier is included since several requests can be running at the same time and the system needs a way to distinguish one request from the other. The result of the query is cached for some time to not overload the Name Server [59].

It is the 16-bit identifier that is used by attackers to insert false records into the DNS cache. An attacker can send numerous malicious UDP-packets which looks like a DNS-response packet, all with different identifiers. One of these might have the same identifier as an open query on the DNS-resolver. If it happens the DNS server will cache the response and return it to every system using it as a resolver.

DNSSEC

The solution for most of the security problems in DNS, such as poisoning, is Domain Name System SECurity Extensions (DNSSEC). The DNSSEC system cryptographically ensures the integrity of the response and makes it nearly impossible to forge a response. If the validation of the integrity fails, the DNS servers return the same error as if the record did not exist. Faults in DNSSEC cannot be temporary accepted by the user as it can with

[4] http://www.verisigninc.com/
[5] http://www.norid.no/
[6] http://www.nic.as/

certificate faults in Hyper Text Transfer Protocol over Secure Sockets Layer (HTTPS). We will give a short introduction to how DNSSEC works. The different elements used is described in Table 2.1.

RR	**Resource Record** RR is an entry in the DNS-system. Depending on the type, it points to resources or data.
RRSIG	**Resource Record Signature** RRSIG is a special resource record containing the signature for one ordinary RR. It is returned at the same time as the RR is validates.
DNSKEY	**Domain Name System Key** DNSKEY is the resource record where the public part of keys is saved, e.g. ZSK and KSK. It also has an accompanying RRSIG.
ZSK	**Zone Signing Key** ZSK is the key used to sign the resource records. It has a *Private* and *Public* part.
KSK	**Key Signing Key** KSK is used to sign the ZSK. It has a *Private* and *Public* part.
DS	**Delegation Signer** DS is a record stored in the parent Zone with a HASH of the public part of the KSK.

Table 2.1: Short descriptions of elements used in DNSSEC.

Every RR in the zone is signed with a private key. The validation process of DNSSEC is quite simple and described in [4, 5]. Each RR in DNS has a RRSIG, which contains the signature for the current RR. The signature can be authenticated with the ZSK, which is a public key stored in a DNSKEY record. The DNSKEY record containing the ZSK is signed with the KSK, available in another DNSKEY record. The record containing the KSK is self signed. To verify this key, a digest of the KSK is stored in a DS record in the parent DNS zone, which in turn has its own RRSIG. The process iterates all the way up to the root. In Figure 2.4 on the following page it is shown which record validates the next. The loop on the right side of the DNSKEY indicates that it is self signed.

The difference between the KSK and the ZSK is that the ZSK is used often, for every DNS RR and every change in the DNS zone. While the KSK only signs the ZSK. It is also reflected in how long these keys are active. The ZSK might be changed once every 2 to 3 months, while the KSK once every year. It would have been possible to just have one key in the DNSSEC-standard. There is several motivations for having two keys: To minimise the number of times an administrator needs to update the key in the parent DNS-server. The KSK can be stronger without impacting the performance (as it only signs a small amount of data). The KSK can have a longer lifetime and can be saved in a more secure place then ZSK [49].

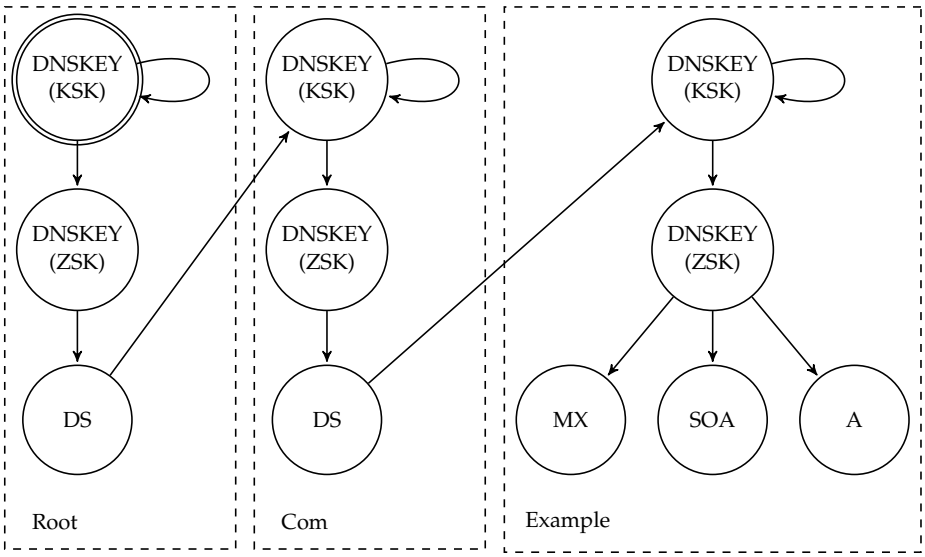

Figure 2.4: DNSSEC signing/validation chain for example.com

In such a system, it is important that the key verifying the root[7] (the DNSKEY with a double line in Figure 2.4) is correct. Root KSK is the public key and trust anchor for the root. It is where the system goes from mathematically provable to depend on trust. A normal user will not have any knowledge about this key and will trust their web browser and local domain name server to have the right key. If a person wants to check the validity of the root KSK, he or she can validate it with the public key of a trusted person that have signed the root KSK. Numerous people have signed the root KSK and can personal attest to its authenticity. Many of them was present when the root key-pair was generated (including personnel from IANA [75]).

It has taken some time for DNSSEC to be implemented and supported in different systems. The DNSSEC Root Key Signing Key was not generated before June 2010.

Some registries have worked to get up the number of domain names with DNSSEC activated. One of these is .SE (top level register for .se domain names [26]). They held a campaign in December 2011 offering cheaper domain names if DNSSEC was enabled, for both new registrations and renewals. Over the year 2011 .SE went from 4 299 domain names with DNSSEC to 171 650 [27]. The Norwegian .no domain will not be signed before the end of 2013. DNSSEC will probably not be available for the holders of .no domains before in 2014.

2.2.2 Secure sockets layer certificate

Secure Sockets Layer (SSL) uses X.509 certificates and are organised in a X.509 Public-Key Infrastructure (PKIX). These are hierarchical structured, however in a slightly different way than the domain name system. The

[7]DNSSEC Root Key Signing Key

certificates are signed by a parent certificate all the way up to the root certificate which is self-signed [17]. For a certificate to be valid, the root certificate has to be pre-installed and a part of the systems PKIX. It is a requirement for the system to be able to validate the chain of trust.

The difference between the domain name system and SSL certificates is that instead of having one clear root, the SSL certificates have multiple roots. It is the software distributor who decides which root certificates would be included in the software. A root certificate are administrated by one Certificate Authority (CA). They negotiate with the software distributor to get their root certificates included and to be a part of the software PKIX.

PKIX supports Certificate Revocation Lists (CRL). These are maintained by the CA and includes all certificates that have been revoked. Most of the certificates in this list is revoked because of administrative reasons, e.g. change in the certificate data. The interesting certificates is those who is on the CRL because of exposure of its private key. As this gives an attacker the opportunity to use the information to set up fake sites.

The functionality that checks these lists is normally deactivated in web browsers and will removed from Google Chrome in the future [30]. The critique against CRL is that it is an old and slow system. It might slow down the process of opening a web page with a second. The CRL also needs to be checked regularly by the browser, so new entries in the list can be recognised.

Security issues

Soghoian and Stamm [70] point out that many government agencies can compel a CA to help in surveillance, by giving the agency a website specific certificate that can be used to spoof a website. The government agency might even get an intermediate CA certificate. This certificate can be used by the agency to make certificates for every site or service they want to, without the CAs knowledge.

A CAs private key can be compromised and used to sign certificates for servers with malicious intent. Recently a CA named DigiNotar was hacked. The attackers used the information to generate a number of certificates. Among the certificates generated was one for the domain name *.google.com [24]. The certificate gave the attackers the opportunity to impersonate different services from Google in browsers who trusted DigiNotar's root certificate.

The identification check of companies done by the CAs may also be inadequate. Someone might get a certificate for a service without having any affiliation to the service. One example of this happened when VeriSign Inc. issued a certificate for "Microsoft Corporation" [55]. The certificate was given to an individual who claimed to be a Microsoft employee.

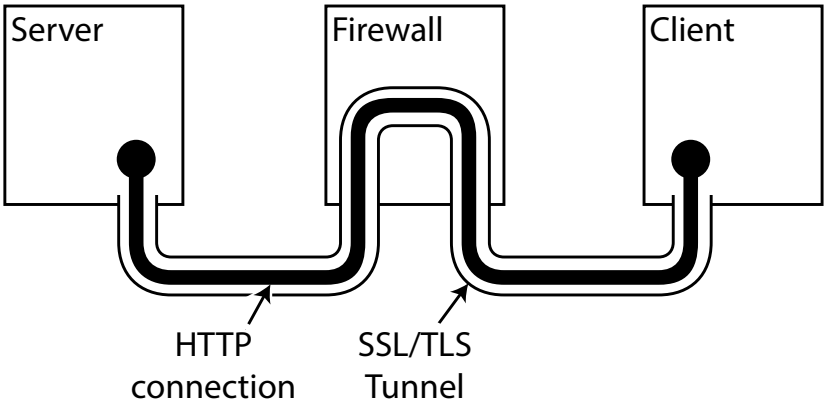

Figure 2.5: Normal firewall without HTTPS Inspection

Firewalls

There is also a rise in the use of firewalls that can inspect application data transferred over SSL. Such solutions has been developed as a countermeasure to use of secure communication for nefarious purposes. It is called a *Trusted man-in-the-middle* [36]. To achieve this the firewall has its own CA. Its public certificate is installed on the devices behind the firewall. When a user opens a web page over SSL, the firewall checks the real certificate of the service and sets up a connection to the service. It then generates a new internal certificate for the connection between the firewall and the client. This certificate is validated since the root certificate of the firewall is present.

Figure 2.5 shows how a normal firewall lets a SSL-tunnel through, while Figure 2.6 on the next page shows how a firewall with HTTPS inspection is an end-point for both the SSL-connections and inspect the content going through the SSL-tunnel. The only way for a user to detect this is to check if the issuer of the certificate is the firewalls CA.

Firewalls that do HTTPS inspection also introduces a single point of failure. If one of the certificates in the firewall is compromised, every device inside the network is at risk without any way to ensure their SSL-connections. Or if an attacker gets access to the firewall, he will be able to see all data sent through it.

2.2.3 DANE

One solution for the problems with SSL certificates is to use DNSSEC and put the certificate in the DNS structure. The DNS-based Authentication of Named Entities (DANE) introduces a new resource record called *TLSA* [37]. This record can contain different types of certificate data, a complete certificate or just a hash of it. It can also have three different kinds of certificates:

1. There is a requirement that one special certificate must be a parent of

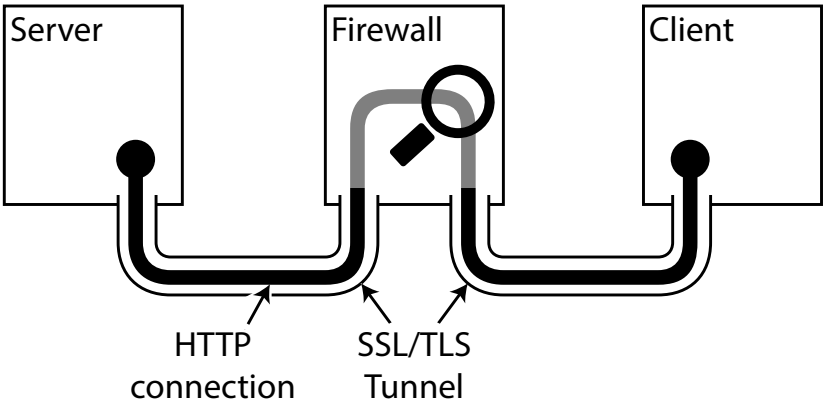

Figure 2.6: Firewall with HTTPS Inspection

the certificate provided by the TLS-handshake. The certificate must also be validated in the services PKIX.

2. The certificate in DNSSEC must be exactly the same as provided by the TLS-handshake. The PKIX must also be validated.

3. The certificate must be exactly the same as provided by the TLS-handshake. This certificate will not be validated against PKIX.

If everyone uses the last alternative it can make CAs obsolete and transfer more power to the administrators of domain name servers. A big advantage will be that the chain of trust will follow the strict hierarchical structure of the DNS, back to the DNS root.

2.2.4 Personalisation

The ability for a user to personalise the content he or she sees when they log on to a website is one way to do *cognitive entity authentication*. The only problem is that most of the solutions available are vulnerable to man-in-the-middle attacks. Such an attack can be done by getting the user to log on to a fake service that works like a proxy to the real service. In this way the attacker can serve the user his or hers personalised information.

Multiple ways has been proposed and implemented to try to authenticate servers. All of them are based on 'Trust-on-first-use'. As the name says, the user has to trust the service on the first use [79]. There is no way to be one hundred percent sure that the service the user is registering to is actually the service he or she wants.

One scenario would be when the user clicks on a hyper link on a page and gets to a social media site. If the user is not familiar with this site, he can be mislead to believe a fraudulent site is the real one and give away personal information.

Several websites let the user choose a security picture[8]. Such a picture will be shown after the user enters his or hers user name and before

[8]Also known as watermarks.

Figure 2.7: An example if a Yahoo login sign-in seal.

entering their password. This solution has also been implemented using text, where the user adds a phrase or sentence for the service to show back to the user when they log on. It can be extended to also include text formatting like color, size and type.

The draft for the Indian National e-Authentication Framework [32] discusses use of such images as a cost efficient alternative to authentication methods that require smart cards or hardware tokens. In their example of a site-to-user authentication system, the authentication process consist of three steps.

1. The user send his or hers user name to the service.
2. The service shows the picture.
3. If it is the correct the user enters his or hers password.

They claim that this has been used for number of years now, by services such as banks etc.

The problem with this solution is that it is vulnerable to a man-in-the-middle attack. Where the user sends his user name to the attacker and the attacker in turn sends this to the service. From the service point of view the attacker is the user and therefore show the security picture. The attacker extracts this picture and shows is to the user. The user would see the picture and recognise it as their own, not suspecting that anything is wrong and enter his or hers password.

Yahoo has used a similar system where a user can select a picture, symbol or words together with a colour scheme. They call this Sign-In Seals [39]. In Figure 2.7 one example is shown where the three words, "Three Secret Words", and colour blue is set by the user. Instead of showing this custom graphics between user name and password, it is shown when the user opens the login page. The graphic in the Sign-In Seal is the same for every user on the same web browser. The page saves a Cookie[9] that is only

[9] A small piece of data saved in the user's browser.

available for pages on that domain name. The data saved in the cookie would not be sent to a different domain, e.g. a phishing site.

The US National Institute of Standards and Technology's SP800-63-1 [11], has some of the same techniques as already described. However they also describes when the personalisation is presented after the login is completed.

The SP800-63-1 publication also discusses email verification, they introduce a method called "*Personalization of email sent to the Subscriber by a valid Verifier*". It works by letting the user select an image when registering to a service. The image is included into every email the user gets from the service provider, giving the user the opportunity to verify the sender. A solution like this would require an attacker to stage a difficult attack, requiring access to either the user mail account, the user information on the service or the communication between these.

Some systems show the user the time and date for the last successful login. It may also include the hostname from where the login originated. The information is often shown to the user after a successful login. The idea is to alert the user about unauthorized usage, or in the case it is missing, the site might be fraudulent. This method is used by the Norwegian E-identity (MinID) and the Norwegian On-line banking identification (BankID).

A big drawback with all such systems was pointed out Schechter *et al.* [68]. They tested how many would still log on to a service even if the expected picture was replaced with a maintanance notice. The result was surprising as 58 out of 60 still entered their credentials. Another test focused on how many would react to a missing SSL indicator. All 63 participants continued to enter their user name and password, which shows how few actually notices such indicators.

2.3 Development of World Wide Web

The Internet has over the years evolved from a set of simple networks used by researchers to communicate with each other, to a complex network delivering services and applications to everybody. The first web browser was just a document reader, now they are capable of run applications that earlier had to be installed on the computer. All these new functionalities and technologies also introduces security challenges.

2.3.1 Dynamic web pages

The big revolution on the web was scripting. The possibility to do complex changes on a web page depending on user input and environment has changed the way developers work. With the development of Asynchronous JavaScript and XML (AJAX) the need to reload a page to get updated information disappeared. Data can be moved freely between the web browser and the web server. It is easier to develop for web, because the solutions do not have to take the operating system nor the type of computer

into account. The deployment of web based systems has also become much simpler, as a user only needs to enter a website address to get access.

2.3.2 Website attacks

Over the years a wide range of different attacks against websites have been invented. Probably one of the most common is *"Denial of service"*-attacks (DoS). These attacks can also be distributed, giving another "D" in the abbreviation. DDoS is hard to combat as they come from multiple sources and can at first glance look like normal traffic. When the generated traffic becomes too high, servers get overloaded and cannot handle the requests from normal users. Which results in the website becoming unavailable.

Cross-Side-Scripting (XSS) is an attack which works by inserting malicious code into a legitimate website through the interactions of the user. Often done by giving a user a link which points to a service. The link includes malicious scripts that are designed to be run on the web page of the service. XSS can have a wide range of purposes, such as transfer of or access to funds, resources or privileges. It is also possible to change the workings of the site, for instance make the page send the user name and password to two servers instead of one.

From our point of view there is one type of attacks that is especially interesting for this project, namely Website cracking[10]. The reason for this is that attackers often attack legitimate websites to hide their phishing site. Website cracking is a generic term for all types of attacks that modifies web pages, complete or just a small part. It has some subcategories, where the last in the list below is our main point of focus.

Defacing
: An attacker just changes the content of a website because he can.

Information acquiring
: Downloading the content, or adding scripts that send information to the attacker.

Virus spreading
: Adding some code that loads hidden pictures, JavaScript, Flash animations or PDF-documents that uses security vulnerabilities to spread malicious code.

Phishing
: Uploading a fake copy of a legitimate service to an already existing website. We discuss this further in Section 2.4 on the next page.

2.3.3 Content Delivery Networks

As more information gets on to the internet, it requires servers that can deliver all the required information to the end user. To be able to do this

[10]Also known as hacking. The informatics community defines often cracking as the malicious form of hacking [28, 29]

each service provider requires a wast amount of servers and bandwidth, which has given rise to Content Delivery Networks (CDN). They place servers on locations all over world, even inside the networks of Internet Service Providers (ISP). This gives the users access to the information fast as the data is available locally. Solutions like this is also lowering the load on the network.

CDNs rent out capacity and storage space to web services, e.g. YouTube and Facebook. Instead of Facebook negotiating with every ISP to be able to place a server in their network, Facebook negotiate with a CDN to put the information in their servers.

Use of such services can be seen in the source code from different service providers. For instance Facebook.com downloads files from a server called *fbstatic-a.akamaihd.net*. The domain *akamaihd.net* is registered to Akamai Technologies, which is a large CDN [76].

2.4 Phishing Attacks

The word *Phishing* was first used in january of 1996, however the attack existed before this [15]. In 1990 Harriman wrote a paper on a related topic using the term *Fishing* [33].

Phishing attacks exploit the weak *cognitive server identification*. It is a way to get a user to give some information about himself to a fake service, while believing this is a legitimate service. The information has mostly been log-in credentials and credit card information, but have evolved into automatic ways to get complete identities for identity thefts.

It is important for the attacker to be careful not to give the user any misgivings. If the user gets suspicious after the attack he or she might change all their passwords and notify credit card issuer or other authorities.

Phishing has always been about profit in some way or another. An attacker can use credentials to get access to resources and information, which again can be sold or used in a way to benefit the attackers cause. In recent years, the selling of identities or user credentials on the black market has become more common.

2.4.1 Types and techniques

There are several types of phishing attacks. The simplest is sending an email and ask for some information. The most used is phishing websites, where a false website gives the impression of being legitimate.

Advanced phishing attacks tries to install some kind of malware on the user's computer. The malware is used by attackers to get information from the computer, e.g. keystrokes and file contents. Malware is outside the scope of this thesis.

Phishing email

Email is the most used form of communication between people on the Internet. Sadly most of the emails sent are characterised as SPAM. A

SPAM-mail can be described as one email sent to many receivers, which the receivers do not want. One of the reasons for its popularity is how simple it is to send an email. Most automatically recognised phishing emails is stopped in SPAM-filters.

In listing 2.1 we show an example of a phishing mail. Here we can see the attacker's use of email addresses to try fool the receiver. They are also writing that the receiver's email can be deleted and the account unavailable if he or she does not answer. The text is made to intimidate the user and get them to provide the requested information. The first problem for this attack to succeed is that the email was sent to a company email to a Norwegian ISP. Which goes to show that this is sent to all e-mail addresses the attacker can get their hands on.

Listing 2.1: Classic example of a phising mail

```
Reply-To: <updatevices@yahoo.co.jp>
From: "IT Services"<ITservices@activist.com>
Subject: New Update
Date: Wed, 30 Jan 2013 10:23:15 -0800

Dear Email User
This message is from Information Technology Services of This EMAIL
to all our Staff. We are currently upgrading our database and
e-mail center and this is our final notification to you.we have
sent several messages to you without response.
We are deleting all unused Mail account to create space for new
accounts. In order not to be suspended, you will have to update
your account by providing the information listed below:
updatevices@yahoo.co.jp

Confirm Your E-Mail Details..
Email.....
User name: .....
Password:.....
Re Confirm Password:.....

If you fail to confirm your continuous usage of our services by
confirming your email password now, your account will be disable
and you will not be able to access your email.

You should immediately reply this email: updatevices@yahoo.co.jp
and enter your password in the above password column.
Thanks for your understanding.
Regard,
IT Services
```

Email as a direct mean to get information like user names and passwords has lost some ground over the years. The reason for this is mostly because people are more cautious about sending sensitive information by email [46]. The scepticism against email can be related to the big number of SPAM emails users on the internet receives every day and the publicity on the topic.

Today, phishing websites are more likely to succeed as users are more accustomed to enter personal information on a web page. Links to a

phishing website can be distributed in emails, posts on social websites, advertisement banners and instant messaging among others [63].

Phishing sites

In [19] Dhamija *et al.* found that a good phishing site was able to fool 90 % of their participants just by using different widely available phishing techniques. These are described in the list below.

Visually deceptive text
: This is character replacement as already discussed in Section 2.1.2 on page 12.

Images masking underlying text
: Attackers can use an image of a legitimate hyper link, which in fact points to another fake site.

Images mimicking windows
: This technique is an image that looks like a real window, but in fact it is a hyper link. The image can look like an error message window, making the user click on the image out of habit.

Windows masking underlying windows
: An attacker can get the browser to spawn new windows. These can be moved to a specific place and show the attackers content.

Deceptive look and feel
: The site is cleverly made and there is only small elements as misspelling or tone of language that give it away. It can also be asking about more information then the site would normally do.

In their set of 22 participants they did not find any correlation between the subjects test scores and their sex, age, education level, the weekly number of hours used on a computer or how acquainted they were with the browser or operating system.

Phishing in the URI

Klevjer [47] described a way to save a complete web page in a link. It was done by using data Uniform Resource Identifier (URI) scheme, where one can set the content type of the media, what kind of encoding has been used and the data itself. He also pointed out that it was possible to save such URIs in Uniform Resource Locator (URL) shortening services e.g. TinyURL.com. Below is one simple example from the paper put in a html link tag.

link

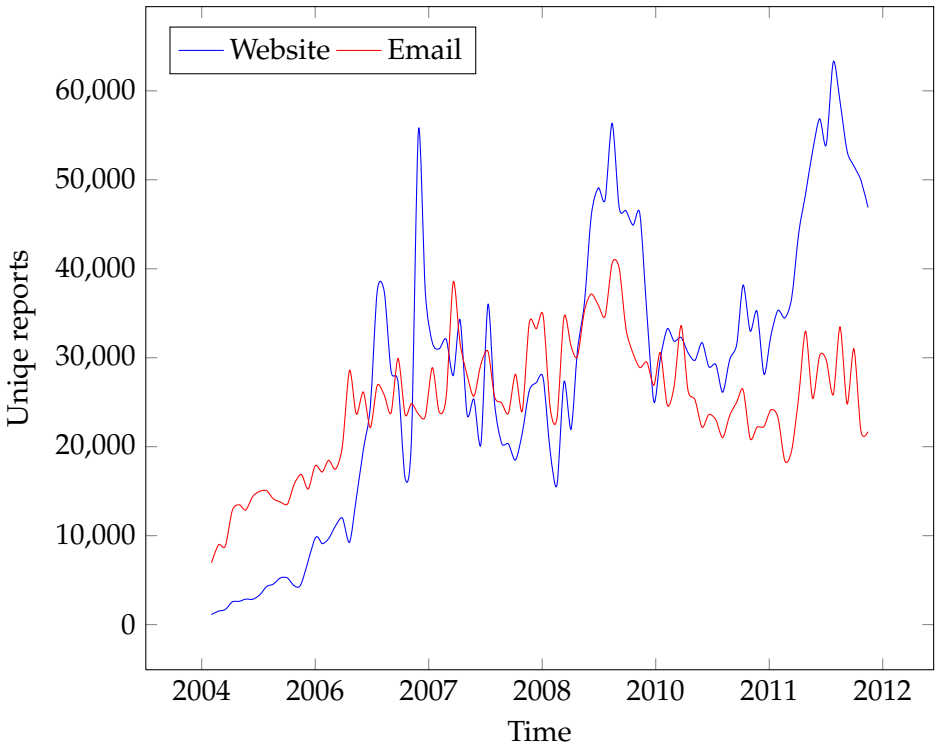

Figure 2.8: Unique phishing reports received by APWG

2.4.2 Statistics

The Anti-Phishing Working Group (APWG) has over the last years collected reports of phishing attempts through emails and websites [1]. The number of unique attempts has been published in their *Phishing Attack Trends Reports*. These numbers have been put together in Figure 2.8. It shows that the number of unique reports varies for phishing websites.

Some of the spikes in Figure 2.8 comes of changes done by the APWG in their methodology. In August 2006 APWG changed the method of counting unique URLs to differentiate between phishing sites on different sub-domains as well as different paths. The change is the reason for the big jump in recorded phishing websites in that same month. The next big spike in April 2007, was caused by multiple phishing sites on the same domain. The next two periods with high number of phishing sites, the second part of 2009 and first part of 2012, is due to higher phishing activity and not to any changes in the method by APWG. It is clear that the number of phishing websites is rising.

The number of phishing emails might be slightly dropping, however it is hard to determine with such short a time span. It is important to note that these numbers are based on reported cases, which means the real number of attempts might be higher.

In APWGs *Phishing Activity Trends Report* from second half of 2011, Carl Leonard from Websense Security Labs states:

"Even fewer phishing web sites are using the oh-so-obvious IP host to host

their fake login pages, instead preferring to host on a compromised domain. There has been a 16 percent drop in the number of phishing URLs containing the spoofed company name in the URL. These combined trends show how phishers are adapting to users becoming more informed and knowledgeable about the traits of a typical phish." [2]

2.4.3 Spear phishing

The term *spear phishing* is used when an attacker has a particular target. Normal phishing attacks try to get credentials from anyone, while spear phishing points out one person or a small group of people. This gives the attacker the opportunity to make the attack so specific that is hard for automatic systems to detect it. One recent example of such an attack was targeted towards some of the executives in the Norwegian telecommunication company Telenor [40]. The attacker got access to their personal computers which includes their Email, files and passwords.

2.5 Petname Model

The Petname Model is a systematic way to personalise global identities. Systems implementing this model allows users to relate identities with some kind of personal media. For instance names, text strings, images or even sounds.

2.5.1 Zooko's triangle

Zooko presented in [82] three desirable properties that a name should have, but cannot have at the same time. Those three properties is known as Zooko's triangle. These properties are *Decentralized*, *Secure* and *Human-Meaningful*. As proposed by Stiegler in [72] we will use "Global" instead of "Decentralized", as "Global" is a more understandable concept. "Memorable" instead of "Human-Meaningful" and "Unique" instead of "Secure". The last renaming might not be as clear as the other two, the reason for the change is that the security of a name lies in its uniqueness.

Zooko's triangle is shown as it is commonly depicted in Figure 2.9 on the following page. A triangle with *Global*, *Unique* and *Memorable* as its corners.

The Domain Name System is probably the closest naming system to incorporate all three properties, however since it is possible for a third-party to register domain names with small changes it is vulnerable to mimicking and thereby phishing. Stiegler points out "In general, phishing depends on mimicry, not forgery" [72].

The properties of names in Zooko's triangle are:

Global
: The name is public and global. It can be exemplified with names of companies, persons or every day objects.

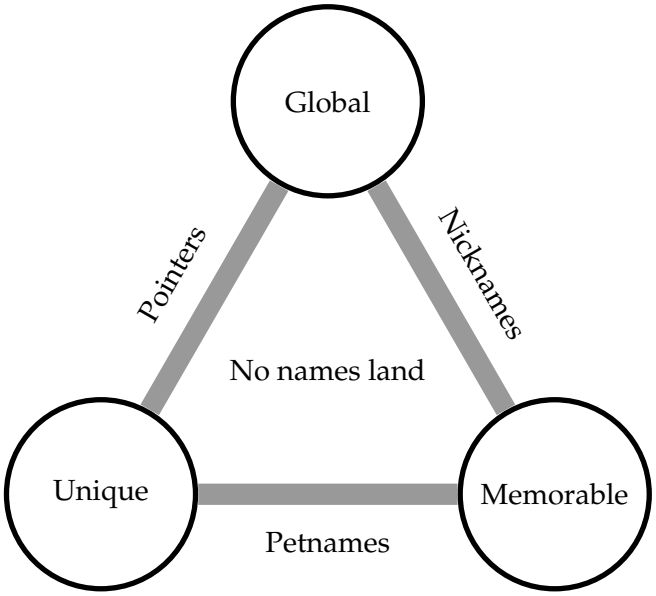

Figure 2.9: Zonko's triangle with elements from a Petname System

Unique
A name is unique syntactically within its domain. For instance phone numbers.

Memorable
A name is easy to remember and recognise. One example often used is the "moving bus test". You see a bus with some name or address on the side and you still remember it when you get home.

When the Petname Model is added into Zooko's triangle, we get different types of names as connections between each property: Pointers, Nicknames and Petnames. These are the connections between the corners in Figure 2.9. The different name types that connects the corners in Zooko's triangle are as follows:

Pointers
They are *Global* and *Unique*, but not *Memorable*. It has also been called "True Name" by Shapiro [69] and "Key" by Miller [58] and Stiegler [72]. The pointer is a unique identifier for one specific person, document, system etc. Social security number combined with nationality is one example of a unique identifier for persons in Norway.

Nicknames
A Nickname is *Global* and *Memorable*, but since a nickname can mean different things to different people, it is not *Unique* in the global domain. A person's given name is such a nickname.

Petnames
Both *Memorable* and *Unique*, but not *Global* as this is personal and

might also be context dependent. Petnames is a chosen name for one person by another person e.g. "dad" or "grandma".

2.5.2 What is the Petname Model?

We need to distinguish between a Petname Model and a Petname System. The model is the idea of a system and its properties, while the system is an implementation of the model.

A Petname Model is a way to relate a personal nickname to a globally known identifier. It lets the user give their important services a personal property, e.g. a name, picture, sound or any combinations of these [72]. One example of a Petname System is a phone book on a mobile phone [22]. When the phone is ringing it shows the name related to the number in the phone book. A smart phone also allow the user to set specific ring- and message tones as well as a picture to a contact.

The same can be done for a list of hostnames, by giving a personal text or image for each website the user visits. An extra function that can be performed in a Petname System is to check its list for similar hostnames and warn the user about a possible phishing attack. It is not in the model itself but would be a useful feature. There might also be possible to extend the system to check cryptographic signatures in the Secure Sockets Layer or other persistent information.

A Petname System can help the user to easily confirm that the service is the same service as he or she already has cognitively authenticated. The user would remember the authentication process that was done when the Petname was created when it is shown. Such a remembrance will help the user to get into the same mindset as when the authentication took place.

2.5.3 Requirements

Ferdous *et al.* describes multiple requirements for a Petname System [22]. These are listed in Appendix B on page 97 and consist of *Functional Properties* (in Table B.1) and *Security Usability Properties* (in Table B.3). The *Functional Properties* consists of requirements to how the Petname System should function. The first is the basis for the system, requiring that a Petname System should at least have one set of *Pointers* and *Petnames*. F2 states that *Nicknames* are optional. For *Pointers* to be resistant against forgery is property F3. The last functional property (F4) is a one-to-one bi-directional relation between the *Pointer* and the *Petname* within each domain. However, F4 has been augmented in a later publication by some of the authors [23]. In the new publication the authors have gone from a strict *bi-directional one-to-one mapping* between the *Petname* and the *Pointer*, to allow a *bi-directional one-to-many mapping* as long as the *Pointer* refers to the same entity (see Table B.2 on page 97). The change is justified by pointing out that one entity can have several pointers and the user should be able to use the same *Petname* for the same entity.

The second category of properties is the *Security Usability Properties* and focuses on the system-user interaction. These can be sorted into two

subcategories *Security Actions* and *Security Conclusions*. The first describes possible actions the user can perform in the system. The other are which conclusions the user can arrive at using the information given by the system.

The *Security Usability Properties* fits into the usability principles for security proposed by Jøsang *et al.* in [44]. These are called *Security Action Usability Principles* and *Security Conclusion Usability Principles*. Both of these has four points as shown in Table B.3 and B.5 in Appendix B on page 97. They are related to one another in a way that the action describes what must be done by the user, and the conclusion describes how the user can assess the security. For instance A2 requires that the user must have the knowledge and ability to make the correct security action. C2 requires that the system provides the information necessary to come to the correct conclusion. A1 and C1 cover the user's understanding of these principles. A3 and C3 describe a tolerable mental and physical load by performing an action or arriving at a conclusion. The two last principles are A4 and C4 they cover the mental and physical load must be tolerable for multiple actions and conclusions.

Some of these properties are relying on each other, e.g. the use of Nicknames. F2 describes that the Nickname is *optional*. If this property is not satisfied by a system, the system is non-compliant with a number of other properties where the Nickname is the main or secondary focus.

2.5.4 Already existing tools

In [22] they evaluated two Petname System Add-Ons for the Mozilla Firefox web browser, called *Petname Tool* and the *TrustBar*. These were evaluated against the properties in Table B.1 and B.3 in Appendix B on page 97. In Table 2.2 on the facing page there is a summery of how these two systems satisfy the properties. Both systems allowed the user to add the same Petname to different pointers, which contradict F4 as there is no longer a one-to-one mapping. This also affect SA7 as the system does not make sure if the new Petname is sufficiently different from existing Petnames. They do not ask the user if he or she would like to add a Petname for highly sensitive data, not meeting the requirement in the SA9 property.

The Petname Tool had some limitations. For example it was not possible to enter nickname (F2) nor did it give the user any Petname suggestions. It did alert the user if a new Petname was resembling an already exiting one.

The TrustBar supported nicknames and provided Petname suggestions based on these nicknames. It did not alert the user if the new Petname resembles the Nickname or an already existing Petname.

As mentioned shortly in Section 1.5.1 on page 4, it has come to our attention that both of these projects are no longer updated and can be considered defunct. *Petname Tool* was last updated on the 30th of June 2009 [16]. The information page for *TrustBar* [35] was last updated late january 2006. It is likely that the development stopped around the same time.

It is hard to determine why these two projects are defunct. It might be

	F				SA									SC				
System	1	2	3	4	1	2	3	4	5	6	7	8	9	1	2	3	4	5
Petname Tool	Y	N	Y	N	Y	Y	Y	N	N	N	N	Y	N	Y	Y	Y	N	Y
TrustBar	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y	N

Table 2.2: Summary of how Petname Tool and TrustBar satisfies the properties from *Ferdous et al.*

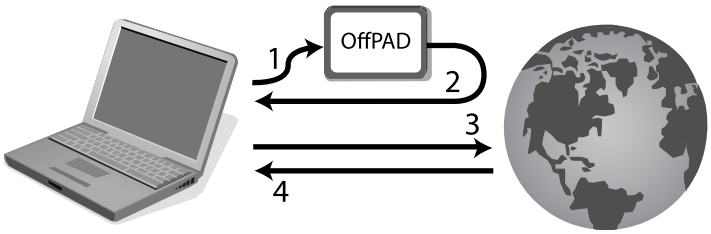

Figure 2.10: How a mobile Petname System is working

because of low number of users. *Petname Tool* is used by 179 users [16]. The reviews of this Add-On shows that it was attractive by users who understood its purpose. The *TrustBar* was discontinued before Add-On manager was released as a part of Firefox in the summer of 2008. There is no information about the number of users for this Add-On.

The most likely reason is that the developers did not have the time nor resources required to keep up with the development of these solutions for new versions of Firefox. It has also been introduced a big number of reputation services for web sites over the last years. Some of these services are extensions to web browsers [7], others as search tools or even built in to the browser [73, 62]. These solutions are being actively updated and may have contributed to the decline in the popularity of the Petname Systems.

2.5.5 Mobile Petname System

Most users are moving between different computing platforms. We access different platforms at work, at the university and at home. The amount of work required to keep one set of Petnames updated on one platform can be considered acceptable. The challenge arises when we introduce two or more sets. It becomes hard to keep track of additions, deletions and other changes that have been done in the different systems. This might even render the browser specific systems unusable.

One solution for this is to make the system mobile, giving the user the freedom to take his set of Petnames with him and use it on every computer he interacts with. Figure 2.10 illustrates with a simple diagram of how this Mobile Petname System works. Where the information about the request is first sent to the OffPAD (1), the OffPAD sends its response back to the computer (2). Depending on the result, the request is sent over the internet (3) and get the requested data in return (4).

2.6 Secure Mobile Devices

Jøsang and Pope describes in [43] a Personal Authentication Device (PAD). It is a secure device external to the computer and is used as an *identity management system*. The user authenticates himself to the PAD using a PIN-code or other available methods. Which in turn starts a time- or connection limited session, in which the PAD acts as the user's identity manager and can automatically authenticate the user to every supporting services. The communication between the PAD and the service is done over a challenge-response protocol through the user's computer. The challenge is changed every time to protect against replay attacks.

Klevjer *et al.* describes in [48] a more secure off-line PAD, an OffPAD. It is a similar device to the PAD except that it should be mostly disconnected. The OffPAD extends the PAD in another way as well. It can manage the identities of service providers and authenticate them.

The reason why the OffPAD should be off-line as much as possible, is to limit the exposure to potential attacks. The limited connectivity is one of the requirements for an OffPAD and can be met by using physically activated (contactless) communication. Other (live connections) means of communications may be appropriate depending on the required assurance level.

If an attacker should get his hands on the OffPAD there must be some kind of access control on the device. The access control may be done by using a PIN, pass phrase, biometrics or other adequate authentication credentials which prevents unauthorized users from accessing the device. It should also include a secure element to prohibit access to the information on the device. The infrastructure for secure messaging and storage in a secure element is described in ISO 7816-4 [38].

Klevjers three requirements for the OffPAD:

- Limited connectivity
- Secure element
- Access control

We also mention in [78] the need for the OffPAD to be tamper resistant. This is to prevent an attacker with physical access to the device being able to alter any of the OffPADs characteristics, e.g. make it possible for the attacker to observe the usage of the device by adding components into it.

Laurie and Singer describe a similar device in [51]. The *Nebuchadnezzar* as they call it, can be used in much the same way as a PAD. They suggest it should run different security applications and give examples of user authentication and transaction signing. The security requirements specified for this device is aimed at the operating system and the functionality. These requirements are quite important and should be considered in the evaluation of a secure device.

The system requirements for the *Nebuchadnezzar*:

- A non-spoofable user interface so the user knows what the device is doing.
- An operating system that is built from the ground up to be secure.
- The device is not to be general purpose (e.g. it doesn't run a web browser).
- It must be able to do cryptography.
- It must be able to do asymmetric cryptography.
- It must interact with the user's untrusted system.
- It must have a user interface.
- It must be updateable.
- It must be able to run multiple applications.
- It must have an absolutely bullet-proof kernel.
- It must be able to that attest to the software is running.

Another personal authentication device is the Pico by Frank Stajano. This device is designed to replace passwords everywhere [71]. Stajano focuses on user authentication in different environments e.g. on web sites, unlocking a screen saver and logging into a computer. Pico requires changes in server and computer applications to be usable.

He suggests a way to unlock the device by using what he calls *Picosiblings*. These siblings can be any type of devices, they just have to support the communication with the Pico. For instance watches, sunglasses and even wigs. If the Pico is in proximity of a number of such devices then the device is in a *friendly environment* and it unlocks.

Stajano describes the problem of getting this solution out into use with a focus on the user. For a user to want to buy such a device, it has to be supported by a wide range of services. To get service providers to support this device it must either be simple to integrate or the number of interested users must be considerable, preferably both. There is no working Pico device at the moment, so it is a long way to go before the technology is ready to be implemented into existing services.

Privacy in regard to an attacker's possibility to identify persons using this technology is not addressed. It is specified that it should not be possible to infer identities or long-term pseudonyms from eavesdropping on communication. Even with a protocol changing the address for the Picosibling at a rapid interval, it is possible for an attacker to gather information about the network of Pico-devices belonging to one person. E.g. who is pico or sibling and number of siblings.

A possible denial of service attack, is to make enough radio interference to break down communication between the Pico and its siblings. Consider the following scenario. In an office building where every employee uses a Pico device, an attacker sends powerful signals on the same frequency

Bluetooth	Medium range point-to-point communication.
NFC	Short point-to-point connections over limited range.
Optical	Using a camera to transfer data, range depending on the resolution and size, normally 5-20 cm.
USB	Wired connection, can also be used to charge the device's battery.
WiFi	Communication over a bigger network, even over the Internet.
ZigBee	Longer point-to-point connections with low power consumption and transmission range up to 100 m [8].
WirelessHART	Industrial quality sensor networks.

Table 2.3: OffPAD communication technologies

used by Pico devices. It will disturb the connections between Pico and its siblings, resulting in lock downs, disturbing every employee in their work. The only way to limit the impact of such an attack would be to use a frequency. Which has to be open for private use and does not travel well through walls and windows. Such attacks might even be a precursor to a spear-phishing attack.

2.6.1 Communication interfaces

The OffPAD needs to connect securely to a computer. It can be done over secure channels or by securing insecure channels. Most of the technologies available for such point-to-point connections are insecure in their nature. We point out in [78] different communication interfaces or technologies the OffPAD can use. These are listed in Table 2.3, including a new optical option.

All these different communication technologies can in theory be used by an OffPAD. The two security concerns that are most important to address are eavesdropping and break-in attempts.

Bluetooth

Bluetooth has some built-in security mechanisms e.g. pairing and pseudo-random frequency hopping. Which makes it difficult for an attacker to listen to the communication, but not impossible. Michael Ossmann [77] made a device capable of listening to Bluetooth communication.

Near Field Communication

Near Field Communication (NFC) builds upon the technology from radio-frequency identification (RFID), where small amounts of data are sent from a RFID tag to a reader. NFC has been developed to support data transfer both ways. Because of the small amount of data, there is no inbuilt security in the standard. The solution is to make a secure channel as proposed in [34]. NFC is quite limited in distance. The entities can only communicate

when they are a few centimetres between them. This physical limitation makes it harder for an attacker to listen to the communication, but it is still possible if a receiver is close enough.

Optical

To use a camera for the communication between two entities is quite easy by using QR-codes or other methods for saving data in graphical elements. It requires a camera and display on both entities to get a two-way communication. A camera is not always available on the users computer, so this is an inflexible method on different platforms.

USB

USB is available on virtually any computer and might be a good fall-back if other technologies are unavailable. To connect the OffPAD to the computer with a cable, takes away the Off in OffPAD. It introduces a possible security vulnerability. As long as the OffPAD is connected, an attacker in the host computer might be able to try a sequence of attacks towards the OffPAD.

WiFi

WiFi introduces some of the same vulnerabilities as with USB. The difference is that USB is point-to-point, while WiFi can be many-to-many. It makes it possible for an attacker to listen to the communication and possibly attack the OffPAD. Even if the WiFi is secured in a number of ways, the user needs to set up the WiFi connection on the OffPAD, which can be a technical challenge and a mental load if it must be done often.

ZigBee

ZigBee is an open standard from the ZigBee Alliance [81]. If NFC is like whispering to each other, ZigBee can be described as standing in a city square and shout. So the problem with eavesdropping is substantial. It is possible to add a "Trust Centre" to manage keys, although in adhoc peer-to-peer communication this is not available. The alternative is to negotiate a link key when setting up the connection, but this key will have to go in the open [80]. ZigBee is simple to set up and have a low power consumption. Communication can be done for a longer moment in time.

WirelessHART

WirelessHART is a wireless version of Highway Addressable Remote Transducer (HART) and is an industrial standard for sending data between smart devices and control- or monitoring system [25]. Smart devices are intelligent field instruments with the capability to save temporary data and route network traffic. It has built-in security, which requires a Security Manager to generate, store, revoke, and renew keys [66]. Even if the

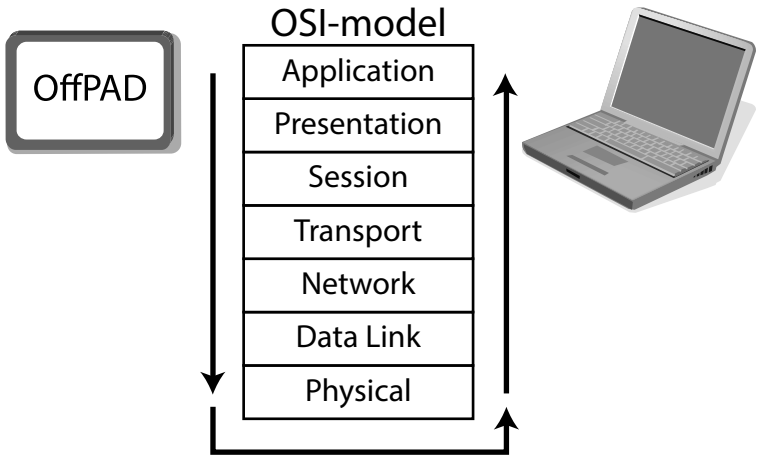

Figure 2.11: The OffPAD and the OSI model

WirelessHART is superior to ZigBee in industrial applications [52], it is not suitable for adhock networks with two communicating parties.

Interface summary

All off these technologies have potential security vulnerabilities. One way to get around this is to encrypt all the data to and from the OffPAD. Using asymmetrical cryptography can give the required security for the connection. In Figure 2.11 there is a diagram of the OSI model. The model can be used as a general model on how a communication system is built. On the bottom we have the *Physical* layer, where the data is electrical pulses, electromagnetic waves etc. The next layer is the *Data Link* layer, where the information on the physical layer is translated to binary data. And so it continues up to the *Application* layer, where the data is used in the intended way. The data between the OffPAD and the application should be encrypted and decrypted on the *Application* layer. This ensures that an attacker in the system, e.g. in a device driver, can not read or alter the data.

In Table 2.4 on the next page, the different technologies are evaluated with the requirements in mind. We can see in this table that the best candidate is NFC. For the management of the device USB would be preferred, as this would be directly connected to the computer and can also be used to charge up the batteries of the OffPAD.

2.6.2 Mobile phone as the OffPAD

Advanced mobile phones, or smart phones, are able to run different user installed applications and must be considered "general purpose computing platform" by the definition in [51]. All the different platforms for smart phones have simple solutions for users to install new applications. It is also easy for developers to make and publish new software to users.

[11] NFC is getting incorporated into more and more devices every day.

Type	Fast set up	Short range	Short connections	Large amount of data	Readily available
Bluetooth				X	X
NFC	X	X	X	X	X[11]
Optical	X	X	X		X
USB	X	X		X	X
WiFi				X	X
ZigBee	X			X	
WirelessHART				X	

Table 2.4: Evaluation of communication technologies

As a consequence it also becomes easy for attackers to make malware applications.

One of the organisations monitoring security, vulnerability and malware against mobile platforms is the Mobile Threat Center (MTC) in Juniper Networks Inc. MTC reported an 155 % increase in malware on mobile phones from 2010 to 2011 [45]. They specified malware on mobile devices to mostly consist of Spyware and Trojans. These are installed as a legitimate applications by the user.

Different operating system manufacturers for mobile phones have a difficult task to combine the request of new technology[12] and keeping their systems secure. Applications are also sandboxed, only getting access to resources the operating system allows. Access to some of the resources often requires user confirmation, e.g. geographical position and contacts.

The operating systems Windows Phone, iOS and Android, all support device management. It gives organisations the opportunity to restrict the functionality of the smart phone and stop the user from installing unwanted and possibly harmful applications. They can also apply settings to the operating system, which includes settings for network and email. Apple [3] and Microsoft [57] has built support for device management[13], while Android uses third-party applications and systems.

On the other hand, the French company TazTag has introduced a mobile phone (TPH-ONE) [74] based on Android. This phone has its own secure hardware element which can be accessed in a secure way. Other phones with a secure hardware elements exists, but this will be the first where it is accessible to the developer of an application. Depending on the implementation, the phone can be regarded as an OffPAD when the secure element is activated.

[12] New technology often introduce new attack vectors and vulnerabilities.

[13] This service might require their own servers

Chapter 3

General Discussion

In this chapter we will consider and evaluate the different technologies and ideas brought up in Chapter 2 on page 9 Background in light of each other. This chapter starts with a short discussions of *Cognitive entity authentication*. We continue with an evaluation of the different technologies used for *Server authentication*, before we briefly touch on the topic of *Phishing*. In Section 3.4 on page 46 we consider the Petname Model and the related requirements. We will also propose several new properties to a Petname System. Lastly we discuss *Secure devices*.

3.1 Cognitive Entity Authentication

Our definition of *Cognitive entity authentication* on page 13 is general and can be used to describe human cognitive authentication of any entity.

One example in everyday life is when person A talks to a random person B on the street. Person A is automatically identified by person B as the entity because he or she can see the other person. Before person B starts communicating, he or she tries to assess the nature of A and if it is acceptable and safe to proceed.

The same process should be done by the user on websites. As persons in real life, websites might be masquerading to be something they are not. For instance there is no guarantee a person in a police uniform is in the police and the website that looks like an on-line bank does not need to be a bank.

In real life, as on the internet, persons can verify claims done by another party. The difference lies in how we recognise a real life object and an internet service. Here we touch on the essence of the problem we try to remedy with the Petname System. We will discuss the Petname Model further in Section 3.4 on page 46

3.2 Server Authentication

We have introduced different ways to authenticate servers. We can sort them into two categories, automatic and manual. The focus in the security

community is more on automatic solutions, e.g. SSL, probably because such technologies is easily managed and general. When an automatic solution is agreed upon it can be implemented in different systems and over time every computer and device will support it. This is not the case for systems in the manual category. They require education and training of every user. The systems must also be simple enough so non-technical persons can use them.

In the first category we find server certificate and DNSSEC, technologies that can without any user interaction determine if the identity of the service is what it claims to be. In the manual category we find technologies that enable the user to do *cognitive entity authentication*, e.g. personalisation and Petname Systems. These require the interaction of a human being, as they check if the service is what the user wants to access.

3.2.1 SSL certificates

There is a large number of CA-certificates in a web browser. If only one of them gets compromised the entire security of the web browser is at risk. The solution could have been the certificate revocation lists (CRL), but because they are slowing down the web browser and the user experience it is not used. The CRL might be a way for an attacker to find certificates that has been compromised.

Security experts agree that traditional server certificates in the browser PKI do not provide adequate protection. The only reason to continue to use them is to encrypt the connection between the client and the server. The certificate do not by themselves ensure confidentiality or integrity as there is no guarantee that the certificate is correct.

3.2.2 DNSSEC

The solution to many of the security problems with the Domain Name System is solved by DNSSEC. A successful DNS poisoning attack (see Section 2.2.1 on page 16) could at the worst be a DoS-attack. It would even stop authorities from guiding visitors from a website over to another by changes performed in DNS.

One of the services affected by this change is the Child Sexual Abuse Anti Distribution Filter (CSAADF), as this is based on locally changed DNS-pointers [50]. At the moment all known domain names to sites distributing child pornography is locally set to point to a web server maintained by the internet service provider. The page that is shown to visitors contains information about why it is shown, definition of child sexual abuse and hyper links to relevant laws. After DNSSEC is introduced the DNS will act like the domain does not exist.

It is only laziness from the domain registries that delays the introduction of DNSSEC. For instance the Norwegian registry NORID has just started discussions on how they should support DNSSEC in their systems. It is not difficult to see that Norwegian Internet Service Providers would

not support DNSSEC before Norwegian domain names are signed with DNSSEC.

Another thing to consider is the trust of the Name Server administrators, as they would be the ones doing the signing.

DANE

Using the DNSSEC to secure the integrity of a service SSL certificate, would be the strongest way to automatically assure the correctness of a certificate. Even if this is not *cognitive entity authentication*, it would form a solid basis for further checks.

When a situation like the one with DigiNotar (as described in Section 2.2.2 on page 19) happens again, services using DANE will not be affected. This depends also on the web browsers ability to validate DNSSEC and DANE.

DANE could also set a stop for firewalls inspecting data going through SSL-tunnels, because the firewall cannot change the information in DNSSEC. From a privacy point of view this is a good solution, as the data would be encrypted all the way from the client to the server. At the same time it removes the possibility for the network administrator to look for malware in the communication. As firewalls inspecting SSL-connection also introduce a single point of failure, it is better to move to other methods of malware detection.

3.2.3 Personalisation

To show the user some elements that they recognize as their own information is a good practice. Personal information is often shown after a user logs on to their on-line bank. Users will be able to see their current balance and maybe a list over their accounts. This is also the case in web mail and on-line forums. It is important to show the personalisation before the user authentication is completed, so the user does not give away all the information to an attacker. It could successfully be defined as a requirement for services using personalisation to show this before the user's password is entered.

Images/Watermarks

Watermarks is a solution that is simple and fast for the user to validate. It is easy for the service provider to support as it would only require an extra field in the table over user data, describing colours or/and strings.

Yahoo's solution is probably the best implementation yet, as their "sign-in seal" is shown before the user enters their password and the data is not available for other sites. The only drawback is that this solution considers one web browser installation and not the user, so the user has to agree with other users of the same computer what this seal should be. It must also configured on each used computer.

Figure 3.1: Third step for logging in with the Norwegian On-line banking identification.

Time of last login

To show the time of last login requires that the user actually remembers when their last login happened. This data is also often available in the user database, which makes it easy to show to the user.

The Norwegian On-line banking identification (BankID) shows the time of last successful login after the user name and the one time password, but before the user enters his or hers personal password. It is the third step and is shown in Figure 3.1. The user can, if vigilant, check if the time is shown and is correct before entering their password.

The Norwegian E-identity (MinID) shows this after the user has given the personal identity number and password, but before entering the one time password. In Figure 3.2 on the facing page it is a screen shot of the form where the user should enter a one time password sent by SMS. Here the user can also see that the last time he or she logged on with these credentials was on the 14th of November 2012. If this is not correct the user can stop the authentication process and contact the authorities to check.

Vulnerabilities

There are mainly two ways to do a phishing attack on sites with personalisation. The first is to send the information through the phishing site to the correct service and return what the user expects to see, i.e. man in the middle. It can be used for *spear phishing*, where the number of users and tries are limited. Many of the service providers do check logs for abnormality and would probably react to many complete or login tries from one IP-address.

The other way is to just remove it and optionally include a message saying the system is being updated. For example if the time of last login

Figure 3.2: Second step for logging in with the Norwegian E-identity.

was missing in Figure 3.2, it would probably have gone unnoticed by most of the users. This is backed up by the research of Schechter *et al.* in [68]. The only way to fool the users of the Sign-In seal (used by Yahoo), is to remove the seal and add a message that the system is unavailable or being updated.

The security minded person would probably notice missing security elements, that everyone else probably would not. It would make personalisation a simple, cheap and a somewhat unreliable service authentication solution.

3.3 Phishing

It is clear from the numbers shown by APWG in Section 2.4 on page 25, that phishing attacks is a massive challenge and will probably continue growing in the future. It is found all over the Internet, in emails, websites, social media and instant messaging.

When it comes to protection against phishing attacks, we have four types of persons:

- Those who are aware of the danger of such attacks that would use the protection mechanisms available.
- Those who are aware of the danger and are of the opinion that they can take care of themselves.
- Those who thinks "this do not happen to me".
- Those who are ignorant of the issue.

The two first would be the most careful using the internet, as they are aware of the danger. The last group can be educated to be aware of the threat. This

type of education and information can be given by banks, employers and other service providers.

The hardest group of people to protect would be those who think it does not happen to them and it would be difficult to convince them otherwise. The most likely reason is that when a person uses the internet, he would notice the poor phishing attacks as phishing attacks. If he comes across a good attack he would not notice it. Providing the person with a somewhat less impression of the real danger and the attitude that he or she can spot all phishing attempts.

3.4 The Petname Model

The Petname Model introduces a way for the user to link their personal relation to a service. It would be hard for an attacker to mimic the service in a Petname System. With the domain name system in mind, in the literature the *Nickname* has been described as optional by Ferdous *et al.* [22] and as the same as *Pointer* by Stiegler [72]. The most logical approach in is the one proposed by Stiegler, as the *Pointer* is the domain name and is normally used as a *Nickname* for a website.

To do such assumptions would not do any change to Zooko's triangle in itself. However it makes it hard to visualise the triangle. The DNS is a naming system that in a small world without malicious intent, would satisfy all three properties; *Global*, *Securely unique* and *Memorable*. Since we have both numerous companies with the same name and people trying to fool users on the internet, the DNS misses on the *Memorable* property. Not in the same way other naming systems, who misses because it is actually hard to remember the name. It is because the DNS is prone to mimicking, where a domain name can with purpose be made to resemble another domain name.

The DNS does not fit the Petname Model completely when we think of the domain name as both a *Pointer* and *Nickname*. As the problem of mimicking is not clearly addressed in the model. If we follow the definition used by Ferdous *et al.*, setting domain name as the *Pointer* and the *Nickname* optional, the DNS will suite the Petname Model better. Although we would lose the *Memorable* property to the domain name which is clearly present. It is easier to put the DNS in to the confines of the Petname Model than the other way around.

Either way the *Petname* as a connection between *Memorable* and *Securely unique* and not *Global*, is not effected by the discussion if a domain name is *Memorable*. A personal *Petname* is ideal to keep track of personal connections to a service.

3.4.1 Petname requirements

The properties for Petname Systems described by Ferdous *et al.* in [22] is made with a different kind of Petname System in mind. They require some amendments to be suitable for an external Petname System. The *Functional*

Properties describe the base for the Petname Model and will be the same for every implementation. It might be interesting to remove the property stating that *Nickname* is optional, as it does not give any higher security and might be easily forged. Especially if it is the title for a web page that would be used as a basis for the *Nickname*.

The same cannot be said for the *Security Usability Properties*, where the *Security Conclusions* describe a system placed into the web browser, giving a new instance for each window and tab. SC1 says *"The Pointer and the corresponding Petname must be displayed at all times through the user interface of the Petname System."*. Such a requirement is difficult for an external device, as it can show requests from different windows and tabs. In Google Chrome it might be possible to determine which window and tab that is active, allowing the Petname System to only react to requests in this tab. It rises a new question, what about the requests done in other windows or tabs. Should these just be ignored or should the Petname System check all request in the same way?

Here we need to consider the development of the content on the internet. All new websites use techniques to make their pages as dynamic and fast as possible. One of the techniques used is background requests, which enables a website to interact with the server without reloading the page. If a site is vulnerable to *cross-site-scripting*, an attacker could insert a script that copies the user name and password and send it to a server after the page loses focus (change of window or tab). Petname System should also be able to control requests done in the background.

Following the argument above a new *Security Conclusion* has to be formed. The goal of SC1 is to make the user confident of the interaction and able to draw the security conclusion easily, to achieve this the Petname System has to check background connections for all open sites. This establishes the basis for NSC1.

NSC1 The user should be informed if a Petname is accessed in a background process.

When opening a web page a number of different servers can be used. For instance when requesting content from *Content Delivery Networks*. Then it is important to not only check what is placed in the address bar of the web browser, but also check every request. The user should be informed if a web page with a specified *Petname* accesses a web server related to another *Petname*. We define the property NSC2 to cover such cases. A web page can also connect to servers which do not have an associated Petname. It can be an indication of an attack and the user must explicitly allow the request to continue. To handle this we define the conclusion property NSC3 and the action property NSA1 on the following page.

NSC2 The user should be informed if a web page with a Petname sends data to a server on another Petname.

NSC3 The user should be alerted if a web page with a Petname sends data to a server without a Petname.

NSA1 It is required by the user to do an explicit action to allow a request from a web page with a Petname to a server without.

3.4.2 Similar pointers

As most of phishing attacks depend upon mimicry of an on-line service, the Petname System should also check for similarities in the pointer to already existing Petnames. If such similarities are found the user should be alerted and required to take an explicit action to make a decision to allow or disallow the request. It raises the question if the action should be saved for future automatic decision making. For example when a user allows a request to a different, but at the same time similar, *Pointer*. It is saved as another pointer to the same entity. It would be useful in cases when the service uses multiple servers, content delivery networks, or both.

It is allowed to add multiple pointers to the same *Petname* in the *functional properties* and therefore should be supported in the *Security Actions* and *Security Conclusions*. So we introduce the two new properties NSA2 and NSC4.

NSA2 The user should be able to add several Pointers to the same Petname.

NSC4 The user should be alerted if a pointer is very similar to an already existing pointer.

3.5 Secure Devices

The Pico device is interesting. However it is too specific to be used with a Petname System. The only focus is to replace passwords and the related infrastructure in computers and websites. The idea of unlocking the device in friendly environments sounds promising. However it needs to be developed with regard to privacy and denial of service attacks. Although Stajano describes that the *picosiblings* can be a large set of personal everyday things, it is unlikely that production of normal clothing, watches and glasses would include such devices. A more likely approach would be to make a range of different size devices that the user could themselves incorporate in to their possessions.

Neither the OffPAD nor the Nebuchadnezzar are complete solutions, i.e. they are ideas for a device not produced yet. The requirements could be a good starting point to make a device specification and designs. Unfortunately this is an expensive process which requires an organisation that is willing to use the device and a large number of them.

3.5.1 Device cost

The cost of an OffPAD is important to consider. A simple one-time-password calculator used by different service providers cost about 8 $[1] a

[1] DIGIPASS GO 3, when ordering 5005 units or more

piece and is very easy to implement in existing systems.

The TazCard which is very similar to what we look for in an OffPAD, was discontinued because of production costs. It requires a large quantity of devices to get the cost per unit down to an acceptable level.

Here we encounter the problem of device cost. It is hard to price a device that does not exist on the market today and what would be an acceptable price for one single device? The cost of a device must be justified by the services it delivers. A company would not buy a device that cost 120 $, when they can get the same service from a device to 8 $.

The Petname System by itself is not the service that would justify an expensive device. However as an application among several others on the same device the price might be considered as acceptable.

3.5.2 Potential users

Banks are likely candidates to support such secure devices, but because of the cost probably only for their high risk customers and employees.

An OffPAD would be interesting for the types of companies who need to ensure the security of their systems. As this device give them the opportunity to make their own applications that fits their needs and requirements. A company might be interested to pay extra for a device like this, since most of the devices available has limited access for developers.

3.5.3 Secure communication

On the first use of the OffPAD with a specific application they need to be paired to each other. In this process it should also exchange cryptographic keys. Prior to the pairing, the OffPAD and the application cannot have any form for secure connection. Since the parring process is based on Trust-On-First-Use (TOFU)[2] it has to be done with some care. It can be a several step process as shown in Figure 3.3 on the next page. It consists of three parts: Securing temporary communication channel, identifying the end-points, and long time key exchange. The first part is a *Diffie-Helman key exchange* [20], which is a temporary key used only for the pairing process. It should be done automatically when a user initialises pairing.

Part two requires the user to manually transfer data from the OffPAD to the application. It is at this point in time that the OffPAD ensures that the communicating party is the application. At the same time that it is the application the user wants to use and not a man-in-the-middle. The process should be a simplified version of the first key exchange, where the application sends its key to the OffPAD and the OffPAD shows its key and a random value to the user. When the key and value is entered into the application, the common key is calculated, the random value is encrypted and transferred to the OffPAD. If the value is correct, the OffPAD does the last part and sends a unique asymmetric key to the application, making the pairing complete. It should be an easy operation for the user to remove the

[2] also known as leap-of-faith

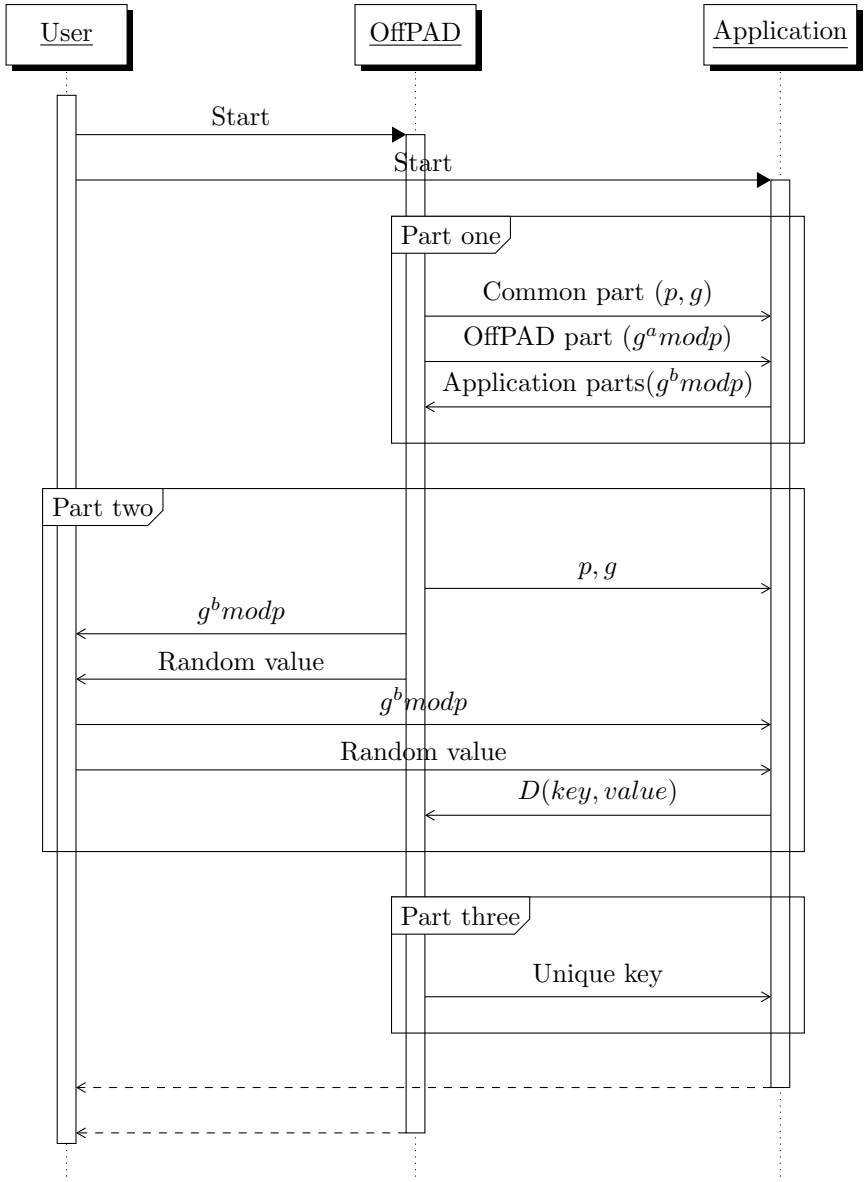

Figure 3.3: Diagram for how the key exchange could be done. The application is in the client terminal.

pairing from either the application (to forget the OffPAD), or the OffPAD (to forget the Application).

Even with an encrypted connection, the OffPAD should remain *offline* as much as possible. All the different technologies that give long connections should be restricted. In fact, the OffPAD should only be connected when the user needs to authenticate and for the management of the device[3]. Since most of the usage of an OffPAD is for short periods of time, it should be easy and fast for the user to set up and tear down the connection. Having a short range of the communication, both physical and virtual, is preferable as this limits the opportunity of eavesdropping and discovery by others. If an attacker knows who you are and when you used your device it can be used as a way to build up trust in a social engineering attack. Another requirement of the technology is that it is able to transfer a large amount of data to and from the OffPAD.

[3]Updating the device, backup and application management (add/remove)

Chapter 4

Technical Description

In this chapter we will describe the technical solution for the complete Petname System. We start with our choices in the design of the system and a description of the device used for the prototype. We continue with a discussion of the implementation of the Petname Model, which also includes an evaluation of our system with regard to the properties discussed in Section 3.4.1 on page 46. This chapter ends with a description of the different components in our Petname System.

4.1 Design Choices

A Petname System can be relatively extensive and difficult to design, so we did some limiting choices to be able to finish in time. The most important will be discussed in this section.

4.1.1 Filtering requests

Our Petname System only considers the domain names to the different websites. It is simple and effective as every domain name will be checked. If the attacker uses a completely unrelated domain name, the user will be asked if he or she wants to add a new Petname for this domain.

To limit the number of request the Petname System has to check, it will only check *POST requests*. This type of requests is normally used to send data to a server. The other type is *GET requests*, which is normally used to get data. A website can have numerous GET requests as most of the items are downloaded this way. By doing this selection it limits the number of checks by the Petname System and there by reducing the number interactions required by the user.

Website developers unfortunately use POST requests to do other functions than just the sending of user data. Any POST-request would be checked by our Petname System, which in turn might require some action from the user. If the number of actions required by the user gets to large it will be a source of mental overload. Resulting in that the user no longer cares about what the system does.

To only check POST requests is a simple solution to a hard problem. It is easy for an attacker to make a site that sends the user name and password in a GET-request and in that way circumvent the Petname check. The reason for not designing a smarter way to determine when the Petname System should check is that it is hard to identify user data among all the information going to and from the server. There is no easy way for web browsers, or site independent systems to determine what the data is or should be. The data can be a user name, a password, a search field or what page to view. It is possible for an attacker to disguise a password so it is impossible to identify it as the user's password.

4.1.2 Connection type

Our prototype will, against the idea of the OffPAD, be connected with the host computer with USB. The reason for this decision is that a standalone Petname System must be running continuously to have the desired effect. If the system was combined with some kind of user authentication it could just be doing service authentication at the same time or just before the OffPAD authenticated the user.

4.1.3 Web browser selection

All the major web browsers have some kind of support for extensions. How this is implemented and developed varies largely in each of them. We needed a web browser where it was simple to develop extensions and at the same time allowed us to do changes in the request handling, e.g. access and block connections.

Internet Explorer was out of the question as it was hard to find any information about how extensions could handle requests. To develop for Safari, Apple requires that the developer register before getting access to development resources. This combined with the small amount of Safari users on windows, stopped us to go further into this web browser.

We wanted to use the Norwegian web browser Opera. Even if it is easy to find information about how to develop extensions for Opera it did not support request handling in extensions. We contacted the developers of Opera just to check if we were missing some information, they suggested we use Google Chrome. We also considered Firefox. It allows extensions to inspect the headers, but Google Chrome was more development friendly.

4.2 System Design

The system can be described by two state machines. One for the browser extension and one for the Petname System. These are strong guides for the rest of development. We will now describe both of them before going into further details of the Petname System.

Figure 4.1 on the next page is the state machine model for the browser extension. Starts by waiting until it gets a *New Request*. In the *Filter* the

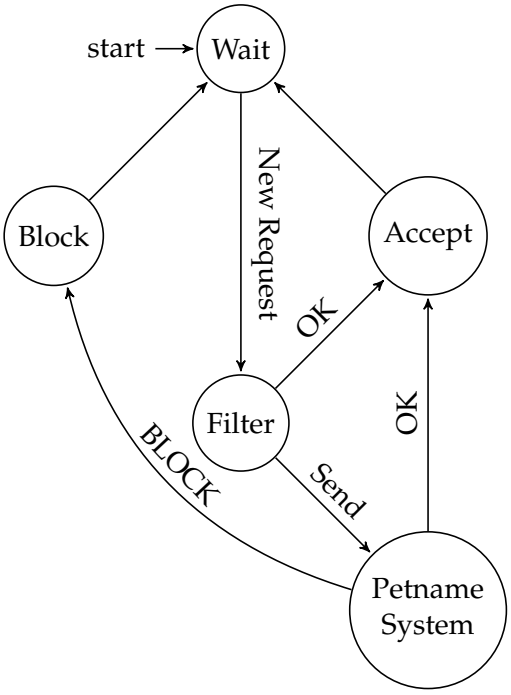

Figure 4.1: State machine for browser extension

GET request gets automatically *OK*, POST requests is sent (*Send*) to the *Petname System* for evaluation. If the *Petname System* returns *OK* the request is accepted, if it returns *BLOCK* the request is blocked. Then it returns to *Wait*.

The workings of the *Petname System* is modelled in Figure 4.2 on the following page. The Petname System waits for a *New query* to be received from the browser extension. When it is received the Petname System performs a *Lookup* in the database. It can give one of three results; *Complete match*, *No/weak match* or *Strong match*.

Complete match The user is *Notified*. At the same time the Petname System send the *OK* message in return to the web browser.

No/weak match The user is asked if he or she wants to add it as a new Petname. Without waiting for an answer the Petname System sends the *OK* message to the web browser.

Strong match The user is shown the domain name of the current request as well as the matching Petname information[1]. The user can choose to let the request continue, which results in that *OK* is returned to the web browser. If the user chooses to stop the request, a *BLOCK* message is returned.

After a message is returned to the web browser the Petname System goes back to *Wait*.

[1]Domain name and Petname.

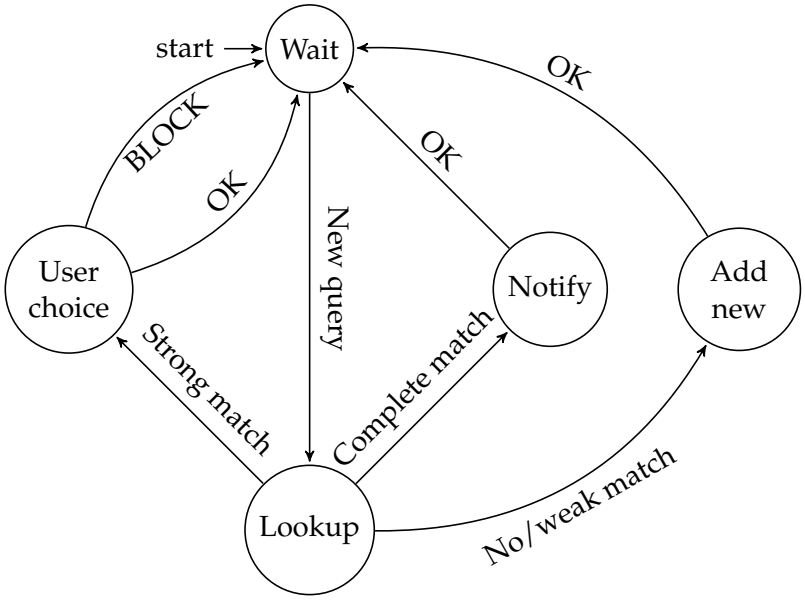

Figure 4.2: State machine for Petname System

4.3 Device

As this master thesis project is a part of the LUCIDMAN-project, we used a product from TazTag, namely the TazCard. It is a small pad (7.0 x 9.8 cm) with a simple Linux distribution and able to run Java Applets, see image in Figure 4.3. It has several communication interfaces: USB, NFC and ZigBee, as well as slots for micro SD card and a smartcard in the same size as a SIM-card.

Figure 4.3: Image of TazCard.

For communication between the computer and the TazCard we used the USB connection, as this was what we had available on the host computer. It also made the project a bit easier to implement, as the USB connection automatically registers itself as a network connection on the host computer. This enabled us to run the Petname server, described in detail in Section 4.5.4 on page 66, directly in the TazCard. If we had to use a ZigBee connection we would have to make a simple proxy server running on the host computer interacting with the device driver.

Regrettably, the production of this device is discontinued due to high production costs. It was the main factor making this project more abstract, giving us the goal to make the system as portable as possible. The TazCard was still used for prototyping and testing as this device is the closest thing we have to an OffPAD.

4.4 Implementation of the Petname Model

The system itself is implemented as one Java class with subclasses to make it easy to include in other projects and onto other devices. The class does not include any user interfaces, as this largely depends on the hardware available and how it should be used in a project. The class is flexible and platform independent and can be included in every user authentication project to add service authentication.

4.4.1 The database class

The database in the Petname System is simple. It can be regarded as a combination of key-value and document database. Key-value databases gives a value in return for the given key and document database has several fields per entry. The key is essential for all the functions in the database, since the key points to a record. A record has three elements; key, title and value. In the prototype, only the key and title fields is actively used. The key is the domain name and the title is the Petname. Values are intended to enable the database to save extra data about a site, e.g. certificate hash, normal login path and so on.

The database has five functionalities, *insert* a record, *delete* an existing record, *update* a record, *select* a record based on key and *list* all records in the system.

Insert

To insert a record, the system makes a new record object where the key title and value is set. When the record is ready to be saved, it is passed to the database which saves it to file.

The record should not have the same key as an already existing record, as during a select the first found record with this key will be returned. Because we check the keys in the user interface and do not give the user the opportunity to add Petnames with the same key, there is no check on duplicate keys at the moment.

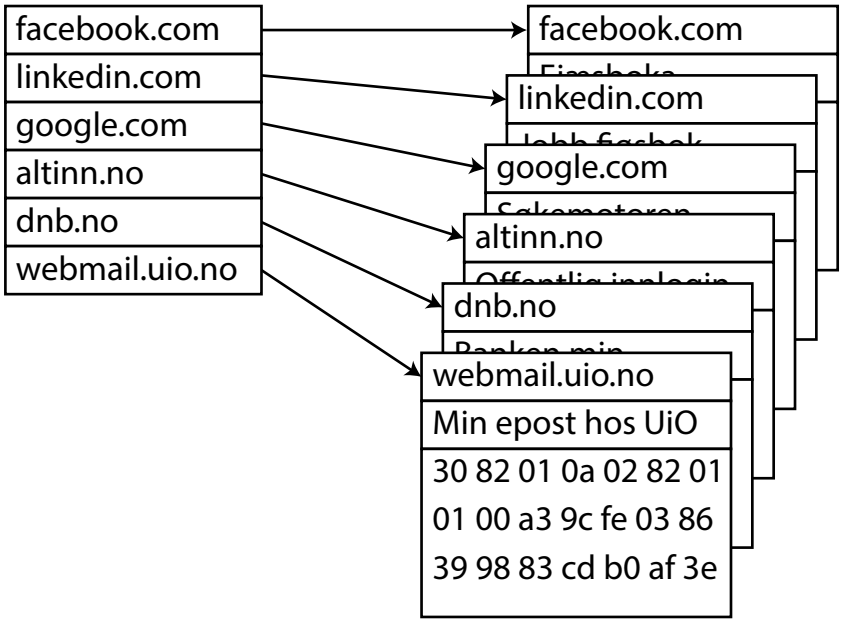

Figure 4.4: A representation of a Key-Value database as implemented in this class.

Delete

There are two ways to delete a record; by key or by object. When using an object the whole record has to be equal before the record is deleted. If only the key is provided, the database finds the record associated with this key and then do the same as when deleted by object. It is more secure to compare the complete records, as the developer can be sure it is the correct record that is deleted. This is done even if the Petname System does not allow a user to add two records with the same key.

Update

When updating a record in the database, one record replaces an old record with the same key. The simplest way to use this function is to first select one record. Do the modifications required on the title or value, and send the record object to the update function.

List All

The *List All* function returns all records as an array. It is useful to keep track of records in the database. The function is used to generate the list of Petnames shown on the settings page in the prototype.

Select

There are two select functions. The first is an exact select. It will only return a record if the correct key is found, otherwise it returns a null value.

Real string	Possible fake string
google.com	g00g1e.com
GOOGLE.COM	G00GLE.COM
UiO.no	Ui0.no
dnb.no	dn6.no

Table 4.1: Examples of similar strings.

The second function uses the string comparison algorithm described in Section 4.4.2. It returns the closest match to the key, if the string comparator returns 20 or lower. It returns a null value if no match is found.

The Petname System first tries to find a record with the same key. If this is successful, it returns the record that matches. Otherwise it uses the select function to find a similar record. If a record is found it is shown to the user as a suspicious site. If no similar record is found, the Petname System ask if it should be added.

Writing and reading data

Because of the lack of documentation for file handling on TazCard, the database file is deleted and fully written every time a record is inserted or updated. Then the database file is reloaded into the database class.

This is not the optimal way to do database file saving. A smarter and more efficient way would be to have a record state bit that determines if the record was deleted. If the record was deleted then the bit could be written to zero. New records could be added to the end of the file.

If a record was updated, the database would first delete the existing record and then insert the modified record as if it was a new record. Once in a while, depending on the amount of space available, the database would do a clean up. Rewriting the whole database and remove deleted records.

4.4.2 String comparison

The heart of the system is the string comparison function. The purpose of this function is to find similar strings based on characters that are optically similar. The similarity depends on the font type being used. Most fonts used in web browsers today have distinct characters. However people are still mistaking the upper case letter "O" for the number zero and the lower case letter "l" for the number one. In Table 4.1 there are some examples of how a real domain name can be faked by an attacker and possibly not noticed by a user.

As there are multiple characters that look quite similar, the function should be able to evaluate the graphical likeness between two characters. By making a predefined lookup matrix where every two characters have an optical relation value, it is possible to make an optical string comparison. In Table 4.2 on the next page a part on the lookup matrix made for this project is shown. Here we can see that the relation value is in the range of zero to nine, where zero is regarded as an equal character and nine is a completely

	A	B	C	D	E	F	G	H	I	J	K	L	1	2	3	4	5	6
A	0	9	9	9	9	9	9	9	9	9	9	9	9	9	9	2	9	9
B	9	0	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
C	9	9	0	9	9	9	3	9	9	9	9	9	9	9	9	9	9	9
D	9	9	9	0	9	9	9	9	9	9	9	9	9	9	9	9	9	9
E	9	9	9	9	0	2	9	9	9	9	9	9	9	9	6	9	9	9
F	9	9	9	9	2	0	9	9	9	9	9	9	9	9	9	9	9	9
G	9	9	3	9	9	9	0	9	9	9	9	9	9	9	9	9	9	2
H	9	9	9	9	9	9	9	0	9	9	9	9	9	9	9	9	9	9
I	9	9	9	9	9	9	9	9	0	9	9	9	1	9	9	9	9	9
J	9	9	9	9	9	9	9	9	9	0	9	9	9	9	9	9	9	9
K	9	9	9	9	9	9	9	9	9	9	0	9	9	9	9	9	9	9
L	9	9	9	9	9	9	9	9	9	9	9	0	1	9	9	9	9	9
1	9	9	9	9	9	9	9	9	1	9	9	1	0	9	9	9	9	9
2	9	9	9	9	9	9	9	9	9	9	9	9	9	0	9	9	9	9
3	9	9	9	9	6	9	9	9	9	9	9	9	9	9	0	9	9	9
4	2	9	9	9	9	9	9	9	9	9	9	9	9	9	9	0	9	9
5	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	0	9
6	9	9	9	9	9	9	2	9	9	9	9	9	9	9	9	9	9	0

Table 4.2: Part of the predefined lookup matrix for characters

different character. The default value for a relation is nine. The lookup matrix is implemented as a two-dimensional array, with two characters as the keys for one single value.

A simplified version of the function takes two strings and checks each character in one string against the character in the same position in the lookup matrix. The value of each lookup is added and multiplied by a hundred, then divided nine times the length of the longest string. The calculation gives a number between one and a hundred which indicates the difference between those two strings. If the returned value is zero means equal strings and a hundred means total different strings.

Listing 4.1 on the facing page shows the pseudo code for the simplified function. On line 2 and 3 it gets the length of the shortest and longest string. It is used in line 4 where the difference in string length is weighted by multiplying the difference with nine. Then each character in these two strings is compared in the lookup matrix in line 8. On line 9, the percentage is calculated and returned.

Listing 4.1: Simplified comparison function

```
function compare(s1, s2):
   minlength := min(length(s1), length(s2))
   maxlength := max(length(s1), length(s2))
   val := (minlength-maxlength)*9
   for i = 0 to minlength:
      char1 := s1[i]
      char2 := s2[i]
      val := val + lookup[char1][char2]
   Return floor((val*100)/(maxlength*9))
```

The complete version also takes the possibility for added or removed characters in the strings. E.g. "petname" and "apetname" will be returned as equal. The offset can be in the middle of the strings, it will make the function to return zero on the two strings "petname" and "petaname". How the function does this is shown in line 18 and 22 in listing 4.2. Where the current char in the first string is compared with the last and the next in second string. The comparison is done three times and the one with the least relation value is chosen.

Listing 4.2: Complete function to compare strings

```
1  public int compare(String st1, String st2) {
2     if (st1.equalsIgnoreCase(st2)) {
3        return 0;
4     }
5     int tmpres;
6
7     char[] s1 = st1.toCharArray();
8     char[] s2 = st2.toCharArray();
9
10    int maxlen = Math.max(s1.length, s2.length);
11    int minlen = Math.min(s1.length, s2.length);
12
13    int maxval = 9 * maxlen;
14    int val = (maxlen-minlen)*9;
15
16    for (int idx = 0; idx < minlen; idx++) {
17       tmpres = 9;
18       if (idx > 0) {
19          tmpres = Math.min(tmpres, lm[s1[idx]][s2[idx-1]]);
20       }
21       tmpres = Math.min(tmpres, lm[s1[idx]][s2[idx]]);
22       if (idx < (minlen-1)) {
23          tmpres = Math.min(tmpres, lm[s1[idx]][s2[idx+1]]);
24       }
25       val += tmpres;
26    }
27
28    return ((val * 100) / maxval);
29 }
```

4.4.3 Validating Petname requirements

When validating the system against the requirements in [22], we will focus on *Functional Properties* and *Security Usability Properties*. All the requirements are listed in Appendix B on page 97.

Security Usability Properties in Table B.4 on page 98 consist of two types of properties, *Security Actions* (prefixed with SA) and *Security Conclusions* (prefixed with SC). The difference between these are mainly that the *Security Action* is something the user does, while *Security Conclusions* is something the user thinks.

Functional Properties

We start with the *Functional Properties* in Table B.1 on page 97. Both F1 and F4 are fulfilled as the pointer has a one-to-one relation to the Petname. Since a domain name can be regarded as a Pointer as well as a Nickname, F2 is also fulfilled. F3 need DNSSEC to be sufficiently met, as the SSL certificate is neither foolproof nor permanent as discussed in Section 2.2.2 on page 18.

Security Action

We will now look closer on how this Petname System meets these *Security Usability Properties* from Table B.4 on page 98 in Appendix B on page 97. As it is the user who needs to add the Petname for the system to recognise it, both *SA1* and *SA2* is satisfied. The user can also edit their Petname for a service in the settings page which is required by *SA3*.

SA4 proposes that the system should be able to suggest the Petname based on the Nickname of the service. It is not implemented, as it is hard to automatically generate Petnames from Nicknames without these being somewhat similar. As there is no support for Petname suggestion there is no need to implement *SA5*, where the user has to accept the suggested Petname with explicit action.

Both *SA6* and *SA7* set requirements to the similarity between names, *SA6* for the Petname and the Nickname and *SA7* for one Petname compared another Petname. It is not activated in our Petname System. However it can easily be done with the string comparison function described in Section 4.4.2 on page 59. The explicit warning required by *SA8* when the user chooses a Petname that is similar to a Nickname or other Petnames, is also not implemented in this system. It is possible to do a similar string comparison between the Petnames and other elements as used to compare keys.

In *SA9* the user should be encouraged to add a Petname for a service that handles highly sensitive data. It is not implemented as it is not possible to automatically infer the sensitivity of the data being handled. It is not a good solution to check if there is a SSL certificate and the complexity of the algorithm. Everyone that wants a SSL certificate can buy one and the grade of its cryptographic complexity is not related to the level of sensitivity of

the data. The system asks the user if he or she would like to add a Petname every time it suspects a user name or password on an unknown service.

Security Conclusion

In this project we choose to show all passing POST-request. It is hard to combine with *SC1* without effecting the user interface in a negative way. *SC1* requires the pointer and the corresponding Petname to be displayed at all times through the user interface of the Petname System. To do this would result in several Petnames showing at the same time, making it hard for a user to notice changes. In a web browser-based Petname System the information can be given in the same place for every tab in the browser.

In this system when a Petname is accessed the screen changes color and the Petname is displayed for 5 seconds, somewhat satisfying *SC2*. It requires the Petname to be clearly visible for the user and grab the user's attention.

For the same reason already mentioned, there is no static indication for missing Petname for the pointer currently in use on the computer. So this Petname System does not satisfy *SC3* directly. However a suspected malicious requests is paused until the user gives his or hers approval.

In normal use Petnames and Nicknames do not show at the same time, fulfilling *SC4*.

The requirement of *SC5* is to clearly warn the user if any of the prior properties is directly violated. In this system every action in the Petname System is clearly shown on the OffPAD, but since it is on an external device it might get out of view.

The new properties

We proposed some new properties in Section 3.4.1 on page 46, as amendments to the *Security Action* and *Security Conclusion properties*. These amendments were developed after our Petname System was finished.

Our Petname System does not consider the owner of the request, so it does not satisfy NSC2 on page 47 and NSC3 on page 47. However it will check requests done in the background, so the system satisfies NSC1 on page 47. Because the user is not warned about the request from a Petname managed site to a non-managed site, he or she can not perform the action required in NSA1 on page 48.

The system does not allow the user to add more than one *Pointer* per *Nickname*, as required by NSA2 on page 48. It was not included as it would require a more complex database. NSC4 on page 48 is satisfied as this system looks for similarities between the used key and existing keys.

Summary

To summarise how our Petname System is compared to the properties in [22] listed in Appendix B on page 97. It is also shown in the beginning of Table 4.3 on the next page. Our Petname System only satisfies half

F				SA									SC					NSA		NSC			
1	2	3	4	1	2	3	4	5	6	7	8	9	1	2	3	4	5	1	2	1	2	3	4
Y	Y	N	Y	Y	Y	Y	N	N	N	N	N	N	N	Y	N	Y	Y	N	N	Y	N	N	Y

Table 4.3: Summary of how our Petname System compares against the properties from [22] and our new properties.

of the requirements, because these requirements is designed with an in-browser Petname System in mind. Where a user can have one instance of the Petname System for each Tab in each web browser process. While in our Petname System has the same instance for every open tab and has to be able to give feedback for each of them at the same time. There are also some of the requirements that is a continuation of optional requirements, e.g. SA4 and SA5.

Our system does not have all the properties we have proposed ourself. It is mostly because these properties were developed after we had seen how our Petname System worked. The only exception was NSC4 on page 48, which has been an idea from the start of this project.

4.5 Prototype

Our prototype incorporates the complete Petname System. It consists of two physical parts; the device and the computer. The device handles most of the processing and interaction happens. While in the computer it is manly the browser extension acting like a gateway between the internal processes in the browser and the device.

4.5.1 Overview

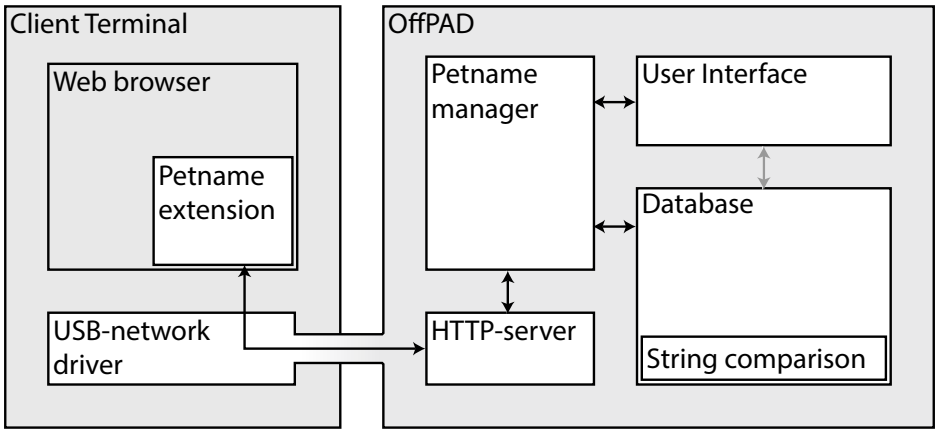

Figure 4.5: Prototype overview.

The prototype consists of six parts as shown in Figure 4.5; The Petname extension, Petname manager, User Interface, Database, HTTP-server and

the USB-network driver. The arrows show how these different parts communicate with each other.

In the web browser it is only the Petname extension that is installed. It works by filtering out the GET-requests and send the rest to the external Petname System, as explained in Section 4.1.1 on page 53. The workings of this extension is described in Section 4.5.5 on page 68. The web browser is not changed in any other way.

The USB-network driver is a standard driver from Microsoft, called Remote Network Driver Interface Specification (NIDS) [56], which can be described as Ethernet over USB. As long as a device reports it has this capability, Microsoft Windows will use this driver. It is developed as a simple way to interface different devices without the need to install a specific driver. As the TazCard supports this, it is easy to program a server on the TazCard to take care of the communication with the host computer and applications. Because of this capability there was no need to make a separate application to run on the computer to interface to the device. Such an application would probably be required if the communication should go over another type of interface, e.g. NFC or ZigBee.

To communicate with the Petname extension in the browser we have a HTTP-server included in the Petname System. It is used as the communication interface. The server is simple and works only with the Petname manager. It is described in detail in Section 4.5.4 on the following page. The Petname manager is the part that manages all the operations of the system. It decides what to show on the user interface depending on the result from the database. The database is an instance of the class already described in Section 4.4.1 on page 57. The user interface is a set of frames and controls to interact with the user.

4.5.2 Petname manager

The Petname manager is the main component of our Petname System. It decides what the user interface should display, what kind of actions should be performed in the database, interact with the HTTP-server. In two cases the Petname manager get commands from the user interface, when the user wants to access the settings page or close the application. When the user accesses the settings page, the Petname manager gives the User Interface the database handle. It is done to simplify the process of deletions and editing of existing Petnames. This is illustrated with the grey arrow between the user interface and database in Figure 4.5 on the facing page.

When a request to check a Petname is received from the Petname extension by the internal HTTP-server. It notifies the manager, which queries the database. Depending on the result from the database, the manager gives what the HTTP-server should respond with. In the case where an explicit action from the user is required, will this response wait on the User Interface. In all the other cases, the response will be sent at the same time as the user is informed.

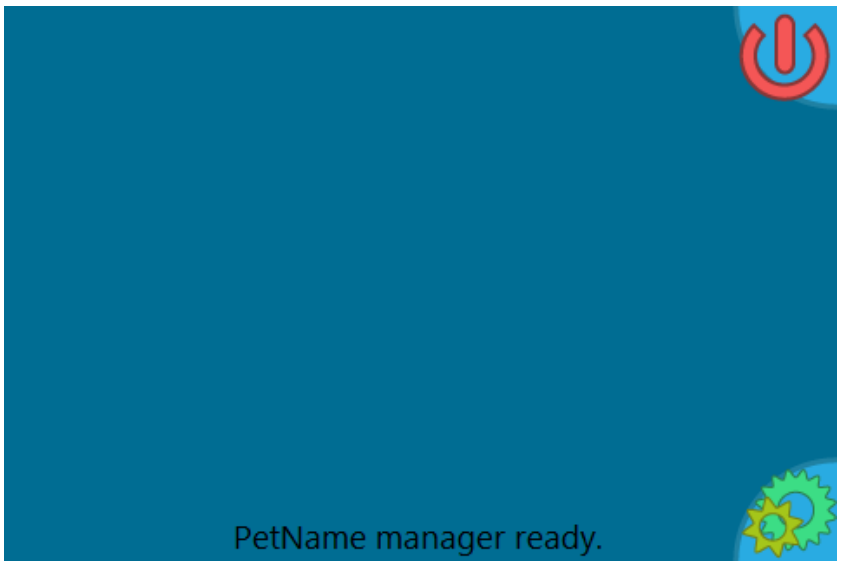

Figure 4.6: Waiting screen on our Petname System.

4.5.3 User interface

The user interface for the prototype is simple by design so not to disturb the user. When the application starts, the "waiting" screen appears, as shown in Figure 4.6. It gives the user the opportunity to go to the settings page to edit or remove existing Petnames.

When the system gets a request to check a domain name, one of three actions can happen depending on the result from the database. If the domain name exists in the Petname System, the Petname is displayed and the background colour is changed for about five seconds. If the domain name is similar to an existing domain name, a warning is displayed where the user can select if he wants to continue or stop the transaction. The last action is where the domain name do not exist nor any similarity is found. The user then gets the opportunity to add this domain name to the Petname System.

As the TazCard did not have any keyboard, we had to develop a way for the user to edit text fields on the device. It was done by drawing a keyboard on the screen with a text box to see what they wrote. The keyboard is shown in Figure 4.7 on the facing page. It has some limitations compared to what is available on smart phones, but it is sufficient.

4.5.4 Device server

To enable communication between the Petname extension in the web browser, a HTTP-server was made. In our prototype this was located in the OffPAD as a part of the Petname System.

The server in the Petname System is working like a very simple HTTP server. One reason for running a HTTP-server instead of a simpler server is because a HTTP request is easily done from a browser extension, and

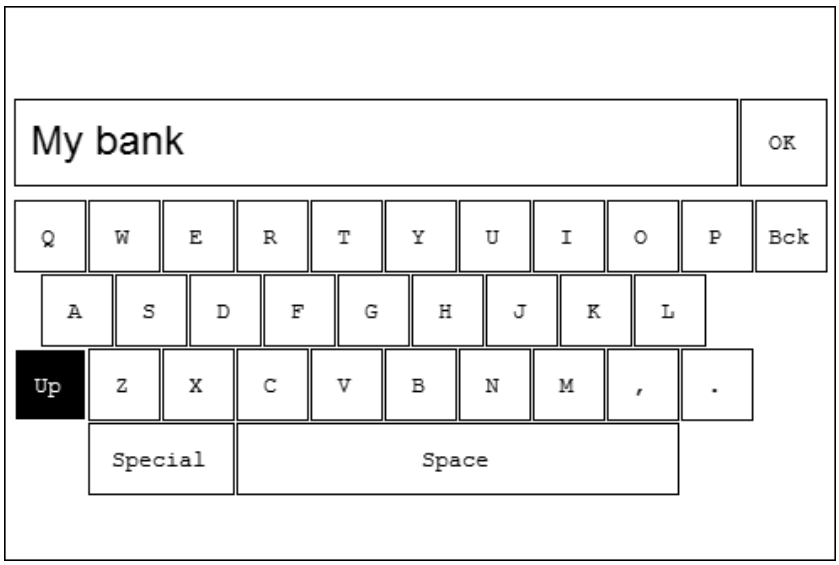

Figure 4.7: Keyboard for the prototype

Listing 4.3: Typical client request

```
1 GET / HTTP/1.1
2 Accept: text/html, application/xhtml+xml, application ...
3 Accept-Charset: ISO-8859-1, utf-8;q=0.7, *;q=0.3
4 Accept-Encoding: gzip, deflate, sdch
5 Accept-Language: en-US, en;q=0.8
6 Cache-Control: max-age=0
7 Connection: keep-alive
8 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) App...
9 X-Petname-Key: webmail.uio.no
```

it is widely supported. The server does not serve any content at the moment e.g. web pages or images. It only response to requests containing a Petname key. There is no problem to extend this service in the future, e.g. if a Petname has an assigned image this can be shown in the web browser.

The client follows the convention[2] of using "X-" in front of its custom experimental HTTP parameters [67]. The Petname key is transferred by the parameter "X-Petname-Key", the content of this parameter is the accessed domain name.

By sending a HTTP-request (see listing 4.3) with the domain name to the server in the OffPAD, the server starts the Petname check. It returns a valid HTTP-response with either "OK" or "BLOCK" like in listing 4.4 on the following page. It can take some time before it is returned, as the Petname manager might wait for an action from the user.

If this system should communicate over NFC or ZigBee, the HTTP-server could run on the computer, which sends the information through the

[2]This convention is not recommended for systems in widely use [67].

Listing 4.4: Typical server anwser

```
1 HTTP/1.0 200 OK
2 Content-Type: text/plain
3 Server: Petname-Controller/0.5b
4 Content-Length: 2
5
6 OK
```

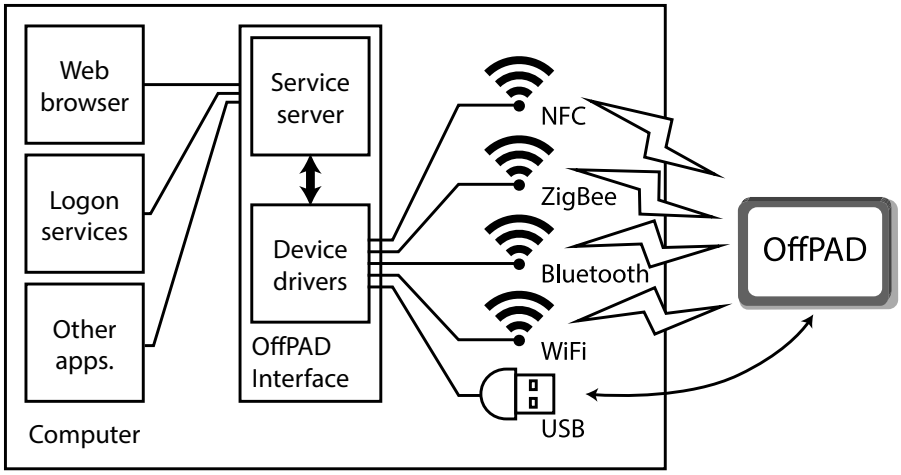

Figure 4.8: Communication with the OffPAD through a computer specific HTTP-server.

computer's device drivers to the OffPAD as shown in Figure 4.8. It would even be possible to send the information over the internet, but then it has to be encrypted and signed to keep the privacy for the user and integrity of the data.

4.5.5 Google Chrome extension

The web browser extension for Google Chrome is based upon the functionality for extensions to block requests to servers. It is mostly used to block advertisements or to block sites with child protection software. The extension registers a listener function with the *chrome.webRequest.onBeforeRequest* handler [31]. Then Chrome will run the function before any request is sent from the browser, and lets the function decide if the request should be stopped or continue.

To limit the number of requests to the Petname-server the extension only checks POST-requests, e.g. login or credit card details, as one web page can issue tens or hundreds of GET-requests. But even with this limitation some websites use POST-requests in scripts that updates parts of a page or to send data about statistics. Such requests will be sent to the

Petname System and might require some user interaction.

When a user sends a POST-request to a site, the extension sends a GET request to the Petname-server like in listing 4.3 on page 67. If the server returns "OK", the POST-request proceed. If it returns "BLOCK", the request is stopped and a message is shown to the user.

4.5.6 Encountered challenges

The challenges encountered in this project was related to the TazCard device.

Backlight problem

In the development of this Petname System we needed to draw the users attention. We tried to get the screen backlight to blink, after it was reduced to zero we could not get it back up to full intensity. It got up to about half intensity. We contacted TazTag support to help us, but they did not find any solution.

As there was other students working on the same device, we were able to borrow a TazCard to do the tests.

Screen locks

Another problem on the TazCard was that the screen stopped responding. After a great amount of debugging the code and rewriting some of the user interface, we found the reason for the fault. The operating system lost contact with the touch sensor on the screen. It is not an error that occurs regularly, so it is hard to find any common denominator.

The way to fix this when it happens is to force a shut down on the device and start it up again. There are some possibilities for the error to still be present after the restart, if so we had to try again.

Chapter 5

User Test

In this chapter we will describe the design and set up of the user test, as well as the result from the test and interviews. Since the type of study impacts every other part of the study, it is one of the first things to be discussed. We continue with the selection of the participants, before the plan and environment of the user tests and interviews. In Section 5.2 on page 76 we present, discuss and analyse the results from the test and interviews.

5.1 Design of the Study

It is important to plan the test and interviews as these can be difficult to get right. There is a number of factors to consider. Such as what kind of study we should perform, how many participants is required and how to select these. The questions asked during the interview need to be carefully formulated to not miss important points nor influence the participants own opinions.

5.1.1 Goal of the study

The test should discover if a Petname System on an external device is something users would use. It should also indicate what people think about phishing attacks with or without a Petname System.

5.1.2 Type of study

Our study can be done as both a qualitative and a quantitative study. Both have their advantages and disadvantages. A qualitative study gives insight, but is hard to analyse. A quantitative study gives statistics and several subjects can answer at the same time. However it can often be taken out of context and it is easy to draw the wrong conclusions.

There are several elements that points to a qualitative study. Some of these are of a practical nature, e.g. we have to use one special computer and a special connected device. It removes the possibility to do several tests at the same time. The most important reason to do a qualitative study is that

it is more valuable for us to get insight into what a subject is thinking than some statistics.

In qualitative study, there is a wide range of different types of methodologies [65], e.g. Conversation analysis, Analytic Induction, Discourse Analysis. We will do a simple case study combined with a usability test, where the user is observed while using the system and interviewed afterwards.

Before we can do a test or interviews where an element of the information we receive can be linked back to the subject, we need to get the permission from the *Data Protection Official for Research*. They are assigned to ensure the privacy of individual persons participating in research studies. In this case it is possible to identify a person from the audio recording of their voice. The form sent to the *Data Protection Official for Research* is available in Appendix E on page 111. We got the approval in just one week, much faster than expected. The letter of approval is included in Appendix F on page 117.

5.1.3 Selection of participants

Nielsen [60] states that a usability tests only needs five subjects. While there is no agreed number of subjects in a qualitative study, experts conclude with "it depends" [9]. Some of the factors to consider are the depth of the interview, what we want to get out of the study and which people are going to be interviewed. None of them are easy to place a number on. There are also some more tangible factors to take into account, like resources and time.

When trying to choose the number of participants, the first factor we considered was the type of people that were required. The target for an external Petname System is a security minded person that uses services on the internet on a daily basis. Then we had to find what we wanted to answer, to see if it would help us settle the number of participants. The main question is simple "Would you use an external Petname System?". As we already know this system could be used and do work, the purpose is to find out if users would use it.

We decided that six participants would be sufficient. It would be more than the required number of five for the user test. The number of answers would also be sufficient to get most of the different opinions about the Petname System presented, as the scope of the questions is limited.

The participants in this test and following interviews are mainly fellow students. Some of them are aware of the topic of our project. However none had any knowledge about the questions or nature of the test. We chose to include two persons that did not have informatics as their field of study, to see if they would have other opinions than the rest.

There is a possibility that their knowledge of the topic might have made them more cautious to the exercises they are going to perform. It might be the same case if the selection of participants is completely random, since the idea of the Petname System and how it works has to be explained to the subject in front of the test anyway.

5.1.4 Questions

The set of questions can be placed into three categories or subsets:

- Describe what has been done.
- Evaluation and thoughts of the system.
- Awareness of phishing attacks.

The first subset is both to check if the subject has understood what he or she tested and to see if the Petname System was easy to use. The second subset of questions is to get the subject's feelings and thoughts around their use of the Petname System. The last subset is just to find out in what degree the subject is aware of possible phishing attacks.

We used the name OffPAD in the questions to refer to the device on the table, and Petname System when addressing the service it provided. The questions for our interviews was as follows:

- While not using the OffPAD
  - Did all web sites work normally?
  - If you noticed anything unusual, what was it?
- While using the OffPAD
  - Did all web sites work normally?
  - If you noticed anything unusual, what was it?
  - Would you have noticed the phishing site without the OffPAD?
  - If you think everything was normal, why did you not notice the warning on the OffPAD?
- General questions to the experiment
  - How did your vigilance change after identifying the first phishing page?
  - How did your sense of security change when using the OffPAD?
  - Were you more or less aware during the experiment then usual?
  - How do you think the use of an OffPAD will impact your daily internet use?
- What would you think about using the Petname System...
  - if it was on your smart-phone?
  - if it was a separate device?
  - if it was on a multi purpose authentication device?
- Normally when accessing and logging on to websites, how aware of phishing attacks are you?

- How do you consider the possibility for phishing attacks against your person?
- What do you think about the usability of the OffPAD prototype?
- Is there something else you want to add?

5.1.5 Interview guide

We developed an *interview guide* for the tests and interviews. Our guide is based on Pathfinder International's guide to designing and conducting interviews [10]. The whole interview guide is available in Norwegian in Appendix D on page 107. A short summary with comments will follow.

The first part is an introduction to this test and interview. Where the subject is informed that the interview is going to be audio recorded. The recording is confidential and the published results will not identify the subject. The subject is also informed that he or she is not obliged to answer any question and can terminate the interview at any moment in time. It is important to respect the person that has given of their time to help you.

The subject gets a short introduction to the Petname System available to them. Then the subject gets four obfuscated links and is asked to enter each of them, log on with a given user name and password and add a Petname for each of them. The user name and password do not have an account so the subject is informed that a "wrong user name and password" message is to be expected. A valid user is not a requirement for the Petname System to work.

Exercise 1
The subject gets four links in random order to each of the sites, where one of them is to a fake site. Then they are asked to log on to each of these, with the Petname System connected.

Exercise 2
The subject gets a new list of four links in random order, where one is to a fake site. It is not the same fake site as in exercise one. The change from exercise one is that the Petname System is deactivated.

After they are finished with these exercises, they will be interviewed about their experience with the question listed in Section 5.1.4 on the previous page. The interview session concludes with an opportunity for the subject to add any last comments or remarks.

The participants are divided into two groups; group A (the first half) and B (the second half). Group A will do exercise one first and then exercise two. While group B will do it in the opposite order.

5.1.6 Phishing sites

There were made four phishing sites just for the use in this test. To keep this simple, the subjects were asked to log in with a non existing user, as

Real domain	Fake domain
nb-no.facebook.com	nb-no.facebook.ccm
accounts.google.com	accounts.google.ccm
www.linkedin.com	www.linkedin.ccm
twitter.com	twitter.ccm

Table 5.1: Used domain names for real and fake sites.

that is enough to trigger the Petname System. There are two reasons for this. The first is that subjects would know the user names and passwords afterwards so we would have to change them all after every interview.

The other reason is that it is easier to make a page that says "wrong user name or password" than to make a page that mimic the content served on the real site. We also did not want the users to enter their own user names and passwords. As it could make the subjects unwilling to participate as they do not have any control over where the data was going.

The phishing sites were composed of a login-page and a "Login failed"-page. The login-page is the same as the landing page for each of the domain names for the sites in question. Only Gmail used the same page for both the login and account refused. Screenshots of the original and the phishing site can be found in Appendix C on page 101.

All the sites used had a .com domain name. It was changed to .ccm (the letter *o* is replaced with the letter *c*) for the fake sites. See Table 5.1 for each of the used domain names. As it is not possible to register .ccm domain names, the domains used in this test was added to the host-file in windows. The host-file is checked for a domain name to IP-address link before windows asks the networks DNS-servers. These domain names pointed to a web server controlled by us.

All the real sites used SSL, our fake pages did not. It was considered if we should add SSL certificates for the fake sites, by making a self signed CA certificate and add it to the web browser. We concluded that it was not necessary, as well as it would be interesting to se if anyone reacted on the lack of SSL.

Under the making of these sites the browser we were using, Google Chrome, reported the fake Facebook site as a phishing site. It is hard to say how Chrome did this. We suspect that one of the images in the page was still pointing back to facebook.com, which resulted in that Google Chrome marked this as suspicious. Most of the sites we made copies of had JavaScript to check if the domain name was the correct and redirect if not. So all the images were saved locally and JavaScripts removed.

We made a change to the web server and the Chrome extension so that only the web browser with this extension got access our phishing sites. The extension was only installed on the computer used for the user tests. It was to ensure no one, including Google, could suspect these sites where real phishing sites. If the server was flagged as a phishing server it would make it impossible for us to use Google Chrome.

5.2 Results

The interviews were done between the 9th and 13th of March 2013, with a total of six participants. As the number of participants are low, the statistics from the test is not representative for any group of people. However their opinions and feedback about the system is still valid. In some cases the number of subjects taking one or the other stand will be mentioned. As it shows if the opinion was a single point of view or if it was more common.

All the recordings from the interviews was deleted as soon they were transcribed and all identifying elements in the transcription removed. The last interview was deleted on the evening of the 13th of March. It is in compliance of the requirements set by the *Data Protection Official for Research* as described in their reply in Appendix F on page 117.

We are not trained interviewers, which might have affected how some of the questions have been asked. These questions can have been formulated in a way that could have lead the subject in their opinion. One example of this is "How did your sense of security increase when using the OffPAD?" where the word "increase" should be "change".

5.2.1 Observations

During the user test it was clear that the system itself was user-friendly. None of the participants had any usability problems with the device. Two encountered the problem mentioned in Section 4.5.6 on page 69 where the screen locks. We had to restart the device to continue the test.

Out of the six persons we interviewed only two did actually check the URL and discovered both the phishing sites before they used the user name and password. There was also one of the participants who saw some error in a graphical element, but when the participant checked the domain name, he or she did not notice the change.

None of the participants reacted to the missing SSL indicator in the web browser. All the real pages used SSL and all the fake ones did not, by observing this a vigilant person is able to circle out the phishing sites. It is as expected when taking into account what we discussed in Section 2.2.4 on page 23 about people not noticing security indicators.

In group B, where they used the Petname System first, all the subjects were more vigilant to possible attacks than the participants in group A. It could be argued that this was against the Hawthorne Effect [54], where a subject is more alert or efficient because they know they are being observed. The set up of this experiment would lead to users being more careful after their discovered the first phishing attack. The test is also too small and simple to say anything about the workings of the Hawthorne Effect.

5.2.2 Findings

The interviewees came with a number of ideas and thoughts under the test about the Petname System, the OffPAD and phishing in general.

We will now go through the points from the interviews and discuss the interviewees comments and feedback.

As mentioned earlier we used the name OffPAD for the device on the table and Petname System for the application. It resulted in that the subjects mostly used the word OffPAD for both. It is of no inconvenience as in this test and interview it is the same thing.

Test without the OffPAD

Four of the subjects noticed a phishing site. The only one in group A said that he or she always checked the URL. In group B two of the subjects found the phishing site because they expected to find one. Both claim in the interview they noticed something they considered to be abnormal, but in fact were completely correct. The first person thought the URL was too long even if it was exactly the same length as the original, the other read *fakebook* instead of *facebook*.

The last person in group B noticed the phishing site without the OffPAD right away. He or she did also pinpoint the phishing site before trying to log into it in the exercise with the OffPAD.

Test with the OffPAD

Every subject reacted on the alert message from the OffPAD when they entered their user name and password into the phishing site. One let the request continue even if he or she noticed the difference in the domain names. When asked why this was done, we got the answer *"it was just to see what happened"*. This illustrates the problem of invoking the curiosity of internet users; they will click.

How vigilance changed

Most of the subjects reported a higher vigilance after the first phishing site was found. Two said that they did not get more vigilant, as they were always vigilant when using the Internet. It was the same subjects that found the phishing site before the OffPAD gave any indication.

As this is a constructed environment and actions, which makes the subject suspicious and likely to expect another phishing attack after the first. How long this heightened vigilance would last in a normal environment is hard to say. We suspect the vigilance would decrease to a normal level in a couple of hours after a failed attack.

Sense of security with the OffPAD

The change in the sense of security between with and without the OffPAD varied from equal to a heighten sense with the OffPAD. Those who had an equal sense of security said it did not add much to what they already did themselves. Or the subjects had certain habits when it came to what websites they usually visits and rarely deviated from these.

The subjects who felt safer described the OffPAD as an extra precaution when surfing the internet.

Impact on daily internet use

None of the subjects mentioned any direct negative implications by using an OffPAD. Most did not think this system would change their behaviour. Here they reiterated the use of web browser bookmarks and writing the address themselves as a reason.

It was also mentioned that the Google search engine was used to find websites on the internet. Google is very efficient to remove suspected phishing and malware sites. They also give the most popular results first, which usually is the correct site.

One of the important points mentioned by one of the subjects was that a Petname System might make him or her indirectly more reckless on the internet. It enlightens a flaw in the implementation. In the way this Petname System is implemented (e.g. not responding to GET-requests) it can give a false sense of security, making them an easier target for phishing attacks.

Devices to use

All the subjects had a negative attitude to a device like the TazCard that only supplied a Petname System. One of the subject would probably use a solution like this for important sites like the Bank and e-Government if it was available, but not for web mail and social networks. Two subjects mentioned that they would have considered it if it was in the size of a key chain. A fourth subject pointed out *"Who would pay for this junk?"*. It is a fair question as such a device might be just as expensive as a simple smart phone.

Most of the participants were positive to a multi purpose authentication device, as this could limit the number of required devices to one. One participant had a different approach to this; he or she said they would be more sceptical to a device with all of their credentials. *"If it got lost you get more problems"*, referring to if someone takes the device and manages to unlock, it they can access every service a user is using. It shows the importance to have a good and simple revocation system in place, which the users have confidence in.

The most preferable solution for all the participants was to use their mobile phone in one way or another. One added *"If it goes seamlessly. Can use some time in the beginning to set it up. (...) As long as you do not need to do any action if the site is already known"*. It shows that a Petname System would be used if it is does not require much user interaction when pages is known and everything is normal.

All the subjects wanted the device to be wireless and not connected with a cable as the TazCard was in this experiment. It should not be necessary to take it out of the pocket or backpack, when it was going to be used.

The threat of phishing

The answers from the participants regarding their awareness to phishing attacks showed that this was not high on their agenda. Most of them did not care or did not believe it would happen to them. The reason was that they mostly used their own bookmarks or typed the domain name themselves when logging in to a site.

There was one exception, where phishing was regarded as a big threat. *"You can loose much if you get phished"*. This person also mentioned indirectly to have been a target for a phishing attack. It was easy to see that this person was more aware, because he or she discovered all the phishing sites before the Petname System could react.

They all considered that the possibility to be a directly targeted in a spear phishing attack as very unlikely.

Usability of the prototype

All the participants had a positive impression of the prototype. The feedback ranged from *"It's all right"* to *"Surprisingly good"*. Size was also mentioned here, it has to be smaller. It also got positive remarks on its response time.

A participant pointed out *"It was intuitive and easy to use (...) for me it was no problems, but if you are not a technical person it might be a bit hard"*. It did not seem to cause any problems for the two non-technical persons participating in this study.

Summary

The key findings from this test and related interviews can be summarised as the following:

- A Petname System can help to discover phishing sites.
- The Petname System did indirectly teach the users what to look for.
- The device used should either be a mobile phone or a small device.
- The user must be aware of the limitations of the Petname System.

Chapter 6

Conclusion

We have in this thesis discussed different methods and technologies that can help to ensure the user's security on the Internet. In the course of this thesis we have answered our research questions from Section 1.4 on page 3.

We have described the challenges of cognitive entity authentication on the internet today, as well as describing a system that protects users from giving away their personal information to unknown and potentially malicious websites. The external Petname System has been developed, tested and evaluated as planned.

6.1 Cognitive Service Authentication

There is absolutely a need for a user to be able to perform *cognitive entity authentication* of service providers. Because more of the technical client-to-service solutions available can be mimicked or even bought by the attacker in a legal way (e.g. SSL certificates). There is also a problem of user awareness about security limitations in systems widely available today. It could be introduced as a system to help users in their assessment of security factors, both as a mean to educate and to secure users.

When the management and operation of a system is placed on the users side, it gets hard for a man-in-the-middle to be able to fool the system with a proxy solution. For same reason such a system will also be superior to most of the solutions of user personalisation described in Section 2.2.4 on page 21.

The Petname System is a proven and working concept. If available, it will help a user to ensure stronger security. As the system focuses on the domain name it might be interesting to combine this with DNSSEC, which uses cryptological functionality to validate the DNS integrity.

Certificates will still be used to ensure a secure channel between the service and the client. It cannot be expected of a user to actually open the certificate to check if it is valid, the mental load will be too high. Just the process for a user to check if the SSL-indicator is present and remember this to the next time he or she accesses the same site, gives a too high mental load for the user. Which can be a part of the explanation for the number of people not noticing the SSL-indicator.

6.2 External Petname System

The External Petname System developed as a part of this thesis, extends other Petname Systems by also checking if an URL is similar to an already known Petname. When evaluating this system against the requirements from [22], it did not come out as good as we would have liked. The reason is that the requirements are made with an in-browser system in mind and not an external system, which is a different environment to handle.

There are some limitations to the developed Petname System. The most important is when it should do the checks, not to overload the user with new Petname requests, and at the same time letting the user add Petnames for the important sites. The ideal case would be to do the check when the user is giving away personal information and ignore when the user just surf the online news services. It is hard to do automatically since it is almost impossible to determine one from the other. Another limitation is that it only supports one kind of web browser and connection only over USB. It should be possible to extend the system to Mozilla Firefox at least. Alternative communication interfaces should be quite simple to develop. It has to be as generic as possible to allow for different kinds of usage.

We used the TazCard for our prototype, which is a cleverly made device. It could have been a perfect solution for an OffPAD. However as the production is stopped we are waiting for other devices that can satisfy the requirements. Although if there is no clear OffPAD solution readily available at the moment, the Petname System can be implemented on smart phones or similar devices. TazTag is currently considering new designs of OffPAD devices in an extension of the Lucidman project where the University of Oslo is a partner.

6.3 User tests

The test subjects were positive to a Petname System and would have used it if it was available. They do not want a new device, they wanted the Petname System in their mobile phone or in a device that would replace another device they already carry. For instance the OTP-calculators from banks. It has to be as small as possible. The size of a keyring was mentioned as optimal.

There was also a question about who should pay for such devices. It depends on who requires the services a device like this can provide. For instance a bank, an employer, an institution or the user themselves. It is important that the cost is as low as possible, to be able to get it into the marked.

Chapter 7

Future work

There is a large number of ways to develop this idea and to use it in different solutions. We will now describe some projects that could follow up on the work done in this theses.

7.1 The Missing Link - User Patterns

More and more websites use third party authentication. One example of this is the student portal (StudentWeb) at the University of Oslo. When you want to log in you get forwarded to a common authenticator (FEIDE). After authentication you get sent back to StudentWeb.

If the Petname Model could be extended to check how the user navigates on the web. It could prevent cross-side-scripting and other malicious code injected on a website. The system can be built up as a graph, giving every web page one node and weight the edge with the number of times the user moves from one page to another. A strong path from the start node to the end node can be regarded as securely unique. Any deviation from the path might be an indication of an attack.

As information only will be saved in the user's own device the privacy preserved.

The biggest challenge in such a system is the amount of data to process and the computing power needed. Both the constructing of the paths and the validation of these has to be done live. The system has to be secure as it will contain much personal information.

7.2 Sign Requests

Data sent between the browser extension and the OffPAD needs to be signed in a way that can be validated on both sides. As the browser is regarded as insecure and there might be a man-in-the-middle between the device driver and the browser extension. It could be done in a similar way as shown in Figure 3.3 on page 50, or by using a kind of service certificate organised in a Public Key Infrastructure.

Such a solution might also include encryption of the data, which would be required if the packet left the computer over an insecure connection, e.g. WLAN and ZigBee.

7.3 Web Browser Extension 2.0

There is a need to make a more intelligent web browser extension. The one made in this project is a simple and rudimentary solution. There is a number of different enhancements that can be done.

- Show in the browser that the user's attention is required on the OffPAD.
- Make support to show images and text from the OffPAD in the browser.
- Let the user pair the browser extension and the OffPAD with a asymmetric key, to ensure privacy and integrity.
- Make a settings page where the user could change the settings for the extension.
- Give the user the opportunity to take and restore a backup of their Petnames through the extension.

7.4 When To Check

One challenge with using a Petname System is when the system should check the database. Opening one of the major news sites in Norway generates over 400 requests. These are requests for pictures, scripts, style sheets and data. These requests are also directed to numerous domain names. If every request was going to be validated by the Petname System the user would have to add tens of Petnames for each site he visits. It is not hard to imagine the user giving up the system in a heart beat.

To combine this with other solutions for user authentication, e.g. external http-digest, it would be a quite small number of requests that would pass through the Petname System.

7.5 Real Life Long Therm User Test

To get a better understanding of how a Petname System would work in daily life, it is necessary to see what people thought about the system after using it for a longer period of time. For instance two or three weeks. After this test period evaluate the usage and the users feelings for this kind of system.

It can with advantage be done simultaneously with other security applications, e.g. validating of DNSSEC and external http-digest.

7.6 Life Cycle

There is a need to define the life cycle for an OffPAD. It should at least cover the following points.

- **Register**
  What has to be done when buying a new device. E.g. generating certificates and register to services.
- **Operate**
  Every day use.
- **Backup**
  How a backup should be taken.
- **Restore**
  How to restore a backup either to current or a new device.
- **Loss**
  What to do if the device is lost.
- **Terminate**
  What has to be done when a device is taken out of service.

It should also contain a plan on how the device can be replaced immediately in the case it is lost or stops working. This includes a revocation of all earlier credentials and the issuing of new credentials for the new device. All applications made for this platform should follow the same life cycle.

7.7 Make a Communication Interface Service

The OffPAD can use several communication interfaces, at the same time the interfaces available on the computer is limited and a browser plug-in could probably only use one.

It requires a service running on a computer that can communicate with the OffPAD. It should be able to do this over different communication interfaces available on the computer, as there is no standardised interface to the OffPAD. The service should also give a single interface for applications on the computer to connect to.

How this communication is done should be standardised and generic to allow multiple computer applications to access multiple devices with their set of services.

Bibliography

[1] Inc Anti-Phishing Working Group. *APWG Phishing Attach Trends Reports*. URL: http://www.antiphishing.org/resources/apwg-reports/ (visited on 09/02/2013).

[2] Inc Anti-Phishing Working Group. *Phishing Attach Trends Report 2nd half 2011*. URL: http://www.apwg.com/reports/apwg_trends_report_h2_2011.pdf (visited on 25/02/2013).

[3] Apple. *Deploying iPhone and iPad - Mobile Device Management*. Sept. 2012. URL: http://images.apple.com/iphone/business/docs/iOS_6_MDM_Sep12.pdf (visited on 09/12/2012).

[4] R. Arends et al. *Protocol Modifications for the DNS Security Extensions*. RFC 4035 (Proposed Standard). Updated by RFCs 4470, 6014. Internet Engineering Task Force, Mar. 2005. URL: http://www.ietf.org/rfc/rfc4035.txt.

[5] R. Arends et al. *Resource Records for the DNS Security Extensions*. RFC 4034 (Proposed Standard). Updated by RFCs 4470, 6014. Internet Engineering Task Force, Mar. 2005. URL: http://www.ietf.org/rfc/rfc4034.txt.

[6] UNINETT Norid AS. *Domain name policy for .no*. 10th Jan. 2012. URL: http://www.norid.no/regelverk/vedlegg-c.en.html (visited on 05/11/2012).

[7] *avast! Internet Security*. URL: http://www.avast.com/en-no/internet-security (visited on 25/02/2013).

[8] N. Baker. "ZigBee and Bluetooth strengths and weaknesses for industrial applications". In: *Computing Control Engineering Journal* 16.2 (Apr. 2005), pp. 20 –25.

[9] "How many qualitative interviews is enough". In: (2012). Ed. by Sarah Elsie Baker and Rosalind Edwards. URL: http://eprints.ncrm.ac.uk/2273/4/how_many_interviews.pdf (visited on 13/03/2013).

[10] Carolyn Boyce and Palena Neale. *Conducting In-depth interviews. A Guide for Designing and Conducting In-depth Interviews for Evaluation Input*. Pathfinder International. May 2006. URL: http://www.pathfind.org/site/DocServer/m_e_tool_series_indepth_interviews.pdf?docID=6301 (visited on 01/03/2013).

[11] William E. Burr et al. *Electronic Authentication Guideline. NIST Special Publication 800-63-1*. English. National Institute of Standards and Technology, Dec. 2011. URL: http : / / csrc . nist . gov / publications / nistpubs/800-63-1/SP-800-63-1.pdf (visited on 14/04/2013).

[12] CoDNS B.V. *About*. URL: http://domain.co.no/gb/about.html (visited on 09/01/2013).

[13] ITU (CCITT). *Information technology - Open systems interconnection - Basic reference model: The basic model*. International Telecommunication Union (formerly known as the International Telegraph and Telephone Consultantive Committee). 1994. URL: http://www.itu.int/rec/T-REC-X.200-199407-I/en (visited on 01/03/201).

[14] ITU (CCITT). *Recommendation X.800, Security architecture for Open Systems Interconnection for CCITT applications*. (X.800 is a re-edition of IS7498-2). International Telecommunication Union (formerly known as the International Telegraph and Telephone Consultantive Committee). 1991. URL: http://www.itu.int/rec/T-REC-X.800-199103-I/en (visited on 25/02/201).

[15] Richard Clayton. "Insecure real-world authentication protocols: or why phishing is so profitable". In: *Proceedings of the 13th international conference on Security protocols*. Cambridge, UK: Springer-Verlag, 2007, pp. 89–96. ISBN: 3-540-77155-7, 978-3-540-77155-5. URL: http : //dl.acm.org/citation.cfm?id=1802438.1802449.

[16] Tyler Close. *Petname Tool*. URL: https : / / addons . mozilla . org / en - US / firefox/addon/petname-tool/ (visited on 04/02/2013).

[17] D. Cooper et al. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280 (Proposed Standard). Internet Engineering Task Force, May 2008. URL: http://www.ietf.org/rfc/rfc5280.txt.

[18] Matt Davis. *Cmabridge*. Cognition and Brain Sciences Unit. URL: http: //www.mrc-cbu.cam.ac.uk/people/matt.davis/cmabridge/ (visited on 05/04/2013).

[19] Rachna Dhamija, J. D. Tygar and Marti Hearst. "Why phishing works". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. CHI '06. Montréal, Québec, Canada: ACM, 2006, pp. 581–590. ISBN: 1-59593-372-7. DOI: 10.1145/1124772.1124861. URL: http://doi.acm.org/10.1145/1124772.1124861.

[20] W. Diffie and M. Hellman. "New directions in cryptography". In: *Information Theory, IEEE Transactions on* 22.6 (Nov. 1976), pp. 644 – 654.

[21] Department of Electronics and Information Technology (DeitY). *e-Pramaan: Framework for e-Authentication*. English. Framework. Version 1.0. Department of Electronics and Information Technology (DeitY), Ministry of Communications & Information Technology, Government of India (GoI), 26th Nov. 2013. URL: http : / / deity . gov .

in/sites/upload_files/dit/files/ePramaan_Framework.pdf (visited on 15/03/2013).

[22] Md. Ferdous et al. "Security Usability of Petname Systems". In: *Identity and Privacy in the Internet Age*. Ed. by Audun Jøsang, Torleiv Maseng and Svein Knapskog. Vol. 5838. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, pp. 44–59. ISBN: 978-3-642-04765-7. URL: http://dx.doi.org/10.1007/978-3-642-04766-4_4.

[23] Md. Sadek Ferdous and Audun Jøsang. "Entity Authenication & Thrust Validation in PKI using Petname Systems". In: *Theory and Practice of Cryptography Solutions for Secure Information Systems (CRYPSIS)* (May 2013). Ed. by Atilla Elçi et al. ISSN: 9781466640306.

[24] Dennis Fisher. *Final Report on DigiNotar Hack Shows Total Compromise of CA Servers*. 31st Oct. 2012. URL: http://threatpost.com/en_us/blogs/final-report-diginotar-hack-shows-total-compromise-ca-servers-103112 (visited on 09/01/2013).

[25] HART Communication Foundation. *What is HART?* URL: http://www.hartcomm.org/protocol/about/aboutprotocol_what.htm (visited on 04/03/2013).

[26] .SE (The Internet Infrastructure Foundation). *About .SE*. URL: https://www.iis.se/english/about-se/ (visited on 09/01/2013).

[27] .SE (The Internet Infrastructure Foundation). *Säkrade domännamn ger säkrare Internet*. 11th Jan. 2012. URL: https://www.iis.se/press/pressmeddelanden/sakrade-domannamn-ger-sakrare-internet/ (visited on 09/01/2013).

[28] geek.com. *Glossary - Cracker Definition*. URL: http://www.geek.com/glossary/C/cracker/ (visited on 29/03/2013).

[29] geek.com. *Glossary - Hacker Definition*. URL: http://www.geek.com/glossary/H/hacker/ (visited on 29/03/2013).

[30] Dan Goodin. *Google to strip Chrome of SSL revocation checking*. Ars Technica. 7th Feb. 2012. URL: http://arstechnica.com/business/2012/02/google-strips-chrome-of-ssl-revocation-checking/ (visited on 31/03/2013).

[31] Google. *chrome.webRequest*. URL: https://developer.chrome.com/extensions/webRequest.html (visited on 01/03/2013).

[32] National e Governance Division. *Draft National e-Authentication Framework (NeAF)*. Version 1.0. Department of Information Technology, Ministry of Communications and Information Technology, Government of India. 1st Sept. 2011. URL: http://www.mit.gov.in/sites/upload_files/dit/files/DraftNeAF1911.pdf (visited on 27/03/2013).

[33] D.D. Harriman. "Password fishing on public terminals". In: *Computer Fraud & Security Bulletin* 1990.1 (1990), pp. 12 –14. ISSN: 0142-0496. DOI: 10.1016/0142-0496(90)90184-M. URL: http://www.sciencedirect.com/science/article/pii/014204969090184M.

[34] Ernst Haselsteiner and Klemens Breitfuß. "Security in near field communication (NFC)". In: *Workshop on RFID Security RFIDSec*. 2006.

[35] Amir Herzberg. *TrustBar: Re-establishing Trust in the Web*. 22nd Jan. 2006. URL: http://u.cs.biu.ac.il/~herzbea/TrustBar/ (visited on 04/02/2013).

[36] Richard Hicks. 18th Oct. 2011.

[37] P. Hoffman and J. Schlyter. *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*. RFC 6698 (Proposed Standard). Internet Engineering Task Force, Aug. 2012. URL: http://www.ietf.org/rfc/rfc6698.txt.

[38] *Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange*. Norm. 2005. URL: http://www.iso.org/iso/iso_catalogue_catalogue_tc/catalogue_detail.htm?csnumber=36134 (visited on 01/04/2013).

[39] Yahoo! Inc. *Give password scams the boot with personalized sign-in seals.* URL: https://protect.login.yahoo.com/ (visited on 04/02/2013).

[40] Per Anders Johansen. *Spionerte på Telenor-sjefer, tømte all e-post og datafiler*. Norwegian. Aftenposten. 17th Mar. 2013. URL: http://www.aftenposten.no/nyheter/Spionerte-pa-Telenor-sjefer_-tomte-all-e-post-og-datafiler-7149813.html (visited on 05/04/2013).

[41] A. Josang et al. "Service provider authentication assurance". In: *Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on*. July 2012, pp. 203 –210. DOI: 10.1109/PST.2012.6297941.

[42] Audun Jøsang. *Identity Management*. 16th Apr. 2012. URL: http://folk.uio.no/josang/im/ (visited on 20/03/2013).

[43] Audun Jøsang and Simon Pope. "User Centric Identity Management". In: *AusCERT Conference 2005*. 2005. URL: http://folk.uio.no/josang/papers/JP2005-AusCERT.pdf (visited on 20/03/2013).

[44] Audun Jøsang, Muhammed Al Zomai and Suriadi Suriadi. "Usability and privacy in identity management architectures". In: *Proceedings of the fifth Australasian symposium on ACSW frontiers - Volume 68*. ACSW '07. Ballarat, Australia: Australian Computer Society, Inc., 2007, pp. 143–152. ISBN: 1-920-68285-X. URL: http://dl.acm.org/citation.cfm?id=1274531.1274548.

[45] Inc. Juniper Networks. *Juniper Mobile Threat Report 2011*. Tech. rep. Juniper Networks, Inc., 2011.

[46] Engin Kirda and Christopher Kruegel. "Protecting Users against Phishing Attacks". In: *The Computer Journal* 49.5 (2006), pp. 554–561. DOI: 10.1093/comjnl/bxh169. eprint: http://comjnl.oxfordjournals.org/content/49/5/554.full.pdf+html. URL: http://comjnl.oxfordjournals.org/content/49/5/554.abstract.

[47] Henning Klevjer. *Phishing by data URI*. 22nd Oct. 2012. URL: http://klevjers.com/papers/phishing.pdf (visited on 20/03/2013).

[48] Henning Klevjer, Kent Are Varmedal and Audun Jøsang. "Extended HTTP Digest Access Authentication". In: *Proceedings of the 3rd IFIP WG 11.6 Working Conference on Policies & Research in Identity Management*. IFIP Springer. London, United Kingdom: Springer, Apr. 2013.

[49] O. Kolkman and R. Gieben. *DNSSEC Operational Practices*. RFC 4641 (Informational). Obsoleted by RFC 6781. Internet Engineering Task Force, Sept. 2006. URL: http://www.ietf.org/rfc/rfc4641.txt.

[50] Kripos. *Internettfilteret CSAADF*. URL: https://tips.kripos.no/cmssite.asp?c=1&h=41&menu=2 (visited on 29/03/2013).

[51] B. Laurie and A. Singer. "Choose the red pill and the blue pill: a position paper". In: *Proceedings of the 2008 workshop on New security paradigms*. ACM. 2009, pp. 127–133.

[52] Tomas Lennvall, Stefan Svensson and Fredrik Hekland. "A comparison of WirelessHART and ZigBee for industrial applications". In: *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*. IEEE. 2008, pp. 85–88.

[53] *LUCIDMAN*. URL: http://www.lucidman.org/ (visited on 18/01/2013).

[54] Rob McCarney et al. "The Hawthorne Effect: a randomised, controlled trial". In: *BMC Medical Research Methodology* 7.1 (2007), p. 30. ISSN: 1471-2288. DOI: 10.1186/1471-2288-7-30. URL: http://www.biomedcentral.com/1471-2288/7/30 (visited on 22/03/2013).

[55] Microsoft. *Microsoft Security Bulletin MS01-017 - Erroneous VeriSign-Issued Digital Certificates Pose Spoofing Hazard*. 22nd Mar. 2001. URL: http://technet.microsoft.com/en-us/security/bulletin/ms01-017 (visited on 09/01/2013).

[56] Microsoft. *USB Remote NDIS Devices and Windows*. 29th June 2009. URL: http://msdn.microsoft.com/en-us/library/windows/hardware/gg463298.aspx (visited on 01/04/2013).

[57] Microsoft. *Windows Phone 8 Device Management Overview*. Oct. 2012. URL: http://go.microsoft.com/fwlink/?LinkId=270085 (visited on 09/12/2012).

[58] M.S. Miller. *Lambda for Humans: The PetName Markup Language*. URL: http://www.eros-os.org/~majordomo/dcms-dev/0036.html (visited on 14/01/2013).

[59] P.V. Mockapetris. *Domain names - implementation and specification*. RFC 1035 (Standard). Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604. Internet Engineering Task Force, Nov. 1987. URL: http://www.ietf.org/rfc/rfc1035.txt.

[60] Jakob Nielsen. *Why You Only Need to Test with 5 Users*. 19th Mar. 2000. URL: http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ (visited on 25/02/2013).

[61] NOMINET. *Second level domains*. URL: http://www.nominet.org.uk/uk-domain-names/about-domain-names/uk-domain-subdomains/second-level-domains (visited on 05/11/2012).

[62] *Norton Safe Web*. URL: http://safeweb.norton.com/about (visited on 25/02/2013).

[63] Gunter Ollmann. *The Phishing Guide*. URL: http://www.technicalinfo.net/papers/Phishing.html (visited on 11/10/2012).

[64] Karen J. Olsen and John M. Tebbutt. *The Impact of the FCC Open Network Architecture on NS/NP Telecommunications Security. NIST Special Publication 800-11*. English. National Institute of Standards and Technology, 1st Feb. 1995. URL: http://www.nist.gov/manuscript-publication-search.cfm?pub_id=890076 (visited on 19/04/2013).

[65] Online QDA Project. *Methodologies*. 29th Nov. 2011. URL: http://onlineqda.hud.ac.uk/methodologies.php (visited on 13/03/2013).

[66] Shahid Raza et al. "Security considerations for the wirelesshart protocol". In: *Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on*. IEEE. 2009, pp. 1–8.

[67] P. Saint-Andre, D. Crocker and M. Nottingham. *Deprecating the "X-" Prefix and Similar Constructs in Application Protocols*. RFC 6648 (Best Current Practice). Internet Engineering Task Force, June 2012. URL: http://www.ietf.org/rfc/rfc6648.txt.

[68] S.E. Schechter et al. "The emperor's new security indicators". In: *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE. 2007, pp. 51–65.

[69] J.S. Shapiro. *Pet names, true names, and nicknames*. 11th Apr. 2000. URL: http://www.eros-os.org/~majordomo/dcms-dev/0036.html (visited on 14/01/2013).

[70] Christopher Soghoian and Sid Stamm. "Certified Lies: Detecting and Defeating Government Interception Attacks against SSL (Short Paper)". In: *Financial Cryptography and Data Security*. Ed. by George Danezis. Vol. 7035. Lecture Notes in Computer Science. 10.1007/978-3-642-27576-0_20. Springer Berlin / Heidelberg, 2012, pp. 250–259. ISBN: 978-3-642-27575-3. URL: http://dx.doi.org/10.1007/978-3-642-27576-0_20.

[71] Frank Stajano. "Pico: No More Passwords!" In: *Security Protocols XIX*. Ed. by Bruce Christianson et al. Vol. 7114. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 49–81. ISBN: 978-3-642-25866-4.

[72] Marc Stiegler. *An Introduction to Petname Systems*. Feb. 2005. URL: http://www.skyhunter.com/marcs/petnames/IntroPetNames.html (visited on 08/09/2012).

[73] *StopBadware*. URL: http://www.stopbadware.org/ (visited on 25/02/2013).

[74] TazTag. *Mobility Products*. URL: http://taztag.com/index.php?option=com_content&view=article&id=104 (visited on 20/11/2012).

[75] Root DNSSEC Design Team. *DNSSEC Trust Anchor Publication for the Root Zone*. 7th May 2010. URL: http://data.iana.org/root-anchors/draft-icann-dnssec-trust-anchor.html (visited on 09/01/2013).

[76] Akamai Technologies. *Akamai Homepage*. URL: http://www.akamai.com/ (visited on 29/03/2013).

[77] Project Ubertooth. *Project Ubertooth*. URL: http://ubertooth.sourceforge.net/ (visited on 01/02/2013).

[78] K. A. Varmedal et al. "The OffPAD: Requirements and Usage". English. In: NSS 2013 (3rd June 2013). Ed. by X. Huang J. Lopez and R. Sandhu. Vol. 7873. Lecture Notes in Computer Science. (to appear). Madrid, Spain: Springer, 3rd June 2013, pp. 80–93.

[79] Dan Wendlandt, David G. Andersen and Adrian Perrig. "Perspectives: improving SSH-style host authentication with multi-path probing". In: *USENIX 2008 Annual Technical Conference on Annual Technical Conference*. ATC'08. Boston, Massachusetts: USENIX Association, 2008, pp. 321–334. URL: http://dl.acm.org/citation.cfm?id=1404014.1404041.

[80] Ender Yüksel, Hanne Riis Nielson and Flemming Nielson. "Zigbee-2007 security essentials". In: *Proc. 13th Nordic Workshop on Secure IT-systems*. 2008, pp. 65–82.

[81] *ZigBee Alliance*. ZigBee Alliance. URL: http://www.zigbee.org/ (visited on 08/04/2013).

[82] Zooko. *Names: Decentralized, Secure, Human-Meaningful: Choose Two*. 22nd Sept. 2003. URL: https://zooko.com/distnames.html (visited on 09/01/2013).

Appendix A

Acronyms

AJAX Asynchronous JavaScript and XML.

APWG Anti-Phishing Working Group.

BankID Norwegian On-line banking identification.

CA Certificate Authority.

CDN Content Delivery Network.

CRL Certificate Revocation Lists.

CSAADF Child Sexual Abuse Anti Distribution Filter.

DANE DNS-based Authentication of Named Entities.

DDoS Distributed Denial of Service.

DNS Domain Name System.

DNSKEY Domain Name System Key.

DNSSEC Domain Name System SECurity Extensions.

DoS Denial of Service.

DS Delegation Signer.

HART Highway Addressable Remote Transducer.

HTTPS Hyper Text Transfer Protocol over Secure Sockets Layer.

IANA Internet Assigned Numbers Authority.

ISP Internet Service Provider.

KSK Key Signing Key.

LUCIDMAN Local User Centric ID Management.

MTC Mobile Threat Center.

NFC Near Field Communication.

NIDS Remote Network Driver Interface Specification.

NIST National Institute of Standards and Technology.

OffPAD Off Personal Authentication Device.

OSI Open Systems Interconnection.

OTP One Time Password.

PAD Personal Authentication Device.

PIN Personal Identification Number.

PKIX X.509 Public-Key Infrastructure.

RFID Radio-Frequency Identification.

RR Resource Record.

RRSIG Resource Record Signature.

SSL Secure Sockets Layer.

TOFU Trust-On-First-Use.

URI Uniform Resource Identifier.

URL Uniform Resource Locator.

USB Universal Serial Bus.

XSS Cross-Side-Scripting.

ZSK Zone Signing Key.

Appendix B

Requirements To Petname Systems

There we have listed the different requirements and properties related to a Petname System.

F1.	A Petname System must consist of at least a Pointer and a Petname.
F2.	Nickname is optional.
F3.	Pointers must be strongly resistant against forgery so that the Pointer can not be used to identify a false entity.
F4.	For every user there must be a bi-directional one-to-one mapping between the Pointer and the Petname of each entity.

Table B.1: Functional Properties [22]

F4.	For every user there must be a bi-directional one-to-many mapping between the Petname and the Pointer of each entity only if these pointers refer to the same entity, otherwise a bidirectional oneto-one mapping between the Petname and Pointer of each entity has to be enforced. That is, the same Petname can be used for different pointers only if all these pointers refer to the same entity.

Table B.2: New Functional Property [23]

A1.	Users must understand which security actions are required of them.
A2.	Users must have sufficient knowledge and the ability to take the correct security action.
A3.	The mental and physical load of a security action must be tolerable.
A4.	The mental and physical load of making repeated security actions for any practical number of instances must be tolerable.

Table B.3: Security Action Usability Principles [44]

SA1.	It is the user who must assign the Petname for each Pointer.
SA2.	Users must assign the Petname for the Pointer with explicit action.
SA3.	As the relationship between the user and other entities evolve, the user should be able to edit the previously applied Petname for a Pointer to a new Petname.
SA4.	Suggestion on the Petname based on the Nickname can be provided as an aid for the user to select a Petname for a Pointer. If the Nickname is missing, other criteria could be chosen for the suggestion.
SA5.	If a suggestion is provided and the user wants to accept it as the Petname, then he must do so with explicit action.
SA6.	Petname Systems must make sure that the user-selected, created or suggested Petname is sufficiently distinct from the Nickname so that the user does not confuse them with each other.
SA7.	Petname Systems must make sure that the user-selected, created or suggested Petname must be sufficiently different from existing Petnames so that the user does not confuse them. This is needed to reduce the risk of mimicry of the Petname upon which the security of the Petname System largely depends.
SA8.	If the user chooses a Petname that may resemble a Nickname or other Petnames, he should be warned explicitly.
SA9.	The User should be alerted to apply a Petname for the entity that involves in highly sensitive data transmission.
SC1.	The Pointer and the corresponding Petname must be displayed at all times through the user interface of the Petname System. This will make the user confident about his interaction and help to draw the security conclusion easily.
SC2.	The Petname for a Pointer should be displayed with enough clarity at the user interface so that it can attract the user's attention easily.
SC3.	The absence of a Petname for a Pointer should be clearly and visually indicated at the user interface so that the user is surely informed about its absence.
SC4.	The visual indication for suggested Petnames and Nicknames should be unambiguous enough so that the user does not confuse them with each other.
SC5.	The warning message that will be provided when there is a direct violation of any of the above properties should be clear enough so that the user can understand the problem and take the necessary security action.

Table B.4: Security Usability Properties [22]

C1.	Users must understand the security conclusion that is required for making an informed decision.
C2.	The system must provide the user with sufficient information for deriving the security conclusion.
C3.	The mental load of deriving the security conclusion must be tolerable.
C4.	The mental load of deriving security conclusions for any practical number of instances must be tolerable.

Table B.5: Security Conclusion Usability Principles [44]

Appendix C

Phishing Sites Compared With The Original

The following screen shoots of web sites on the next pages. The fake sites had only one function; ask for user name and password and return invalid password. They were also as like as possible to the original. The work done is nothing more then what any attacker could replicate.

C.1 Facebook

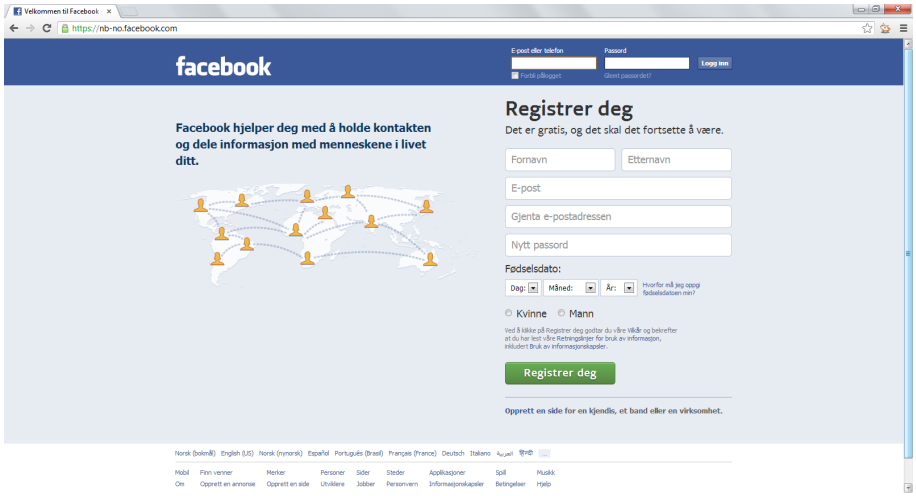

Figure C.1: Real Facebook site

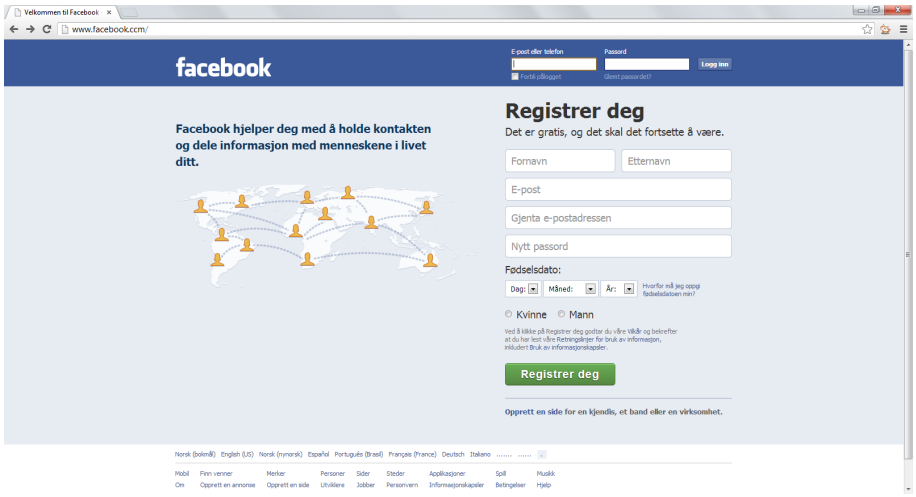

Figure C.2: Fake Facebook site

C.2 Gmail

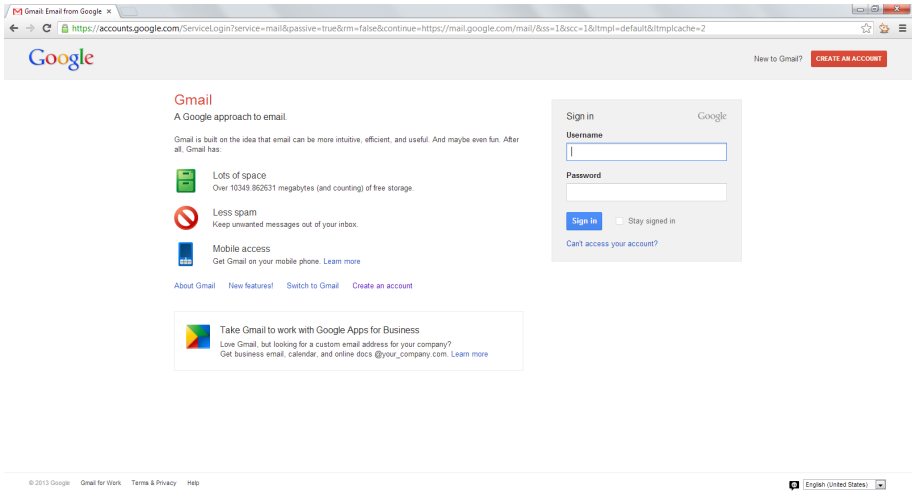

Figure C.3: Real Google site

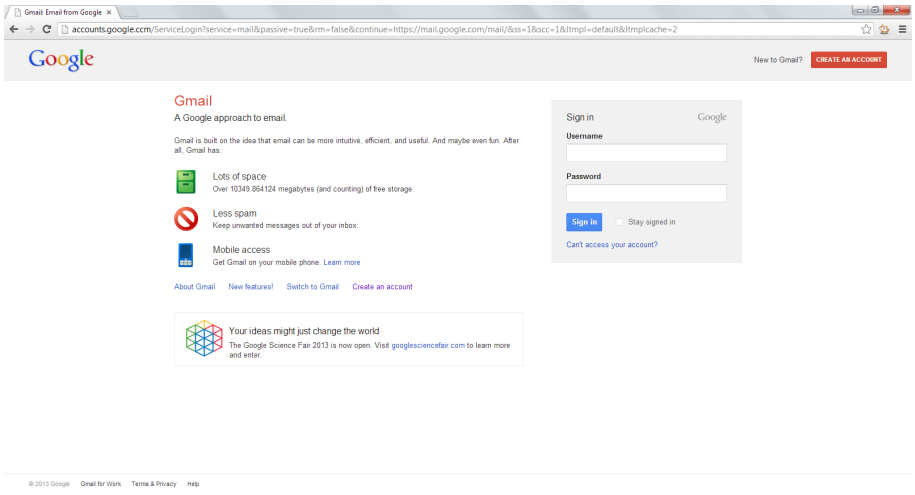

Figure C.4: Fake Google site

C.3 Linkedin

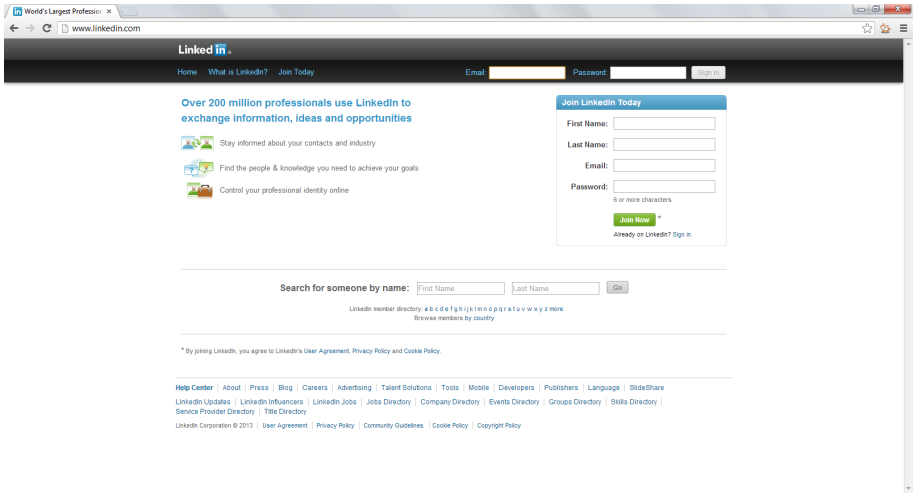

Figure C.5: Real Linkedin site

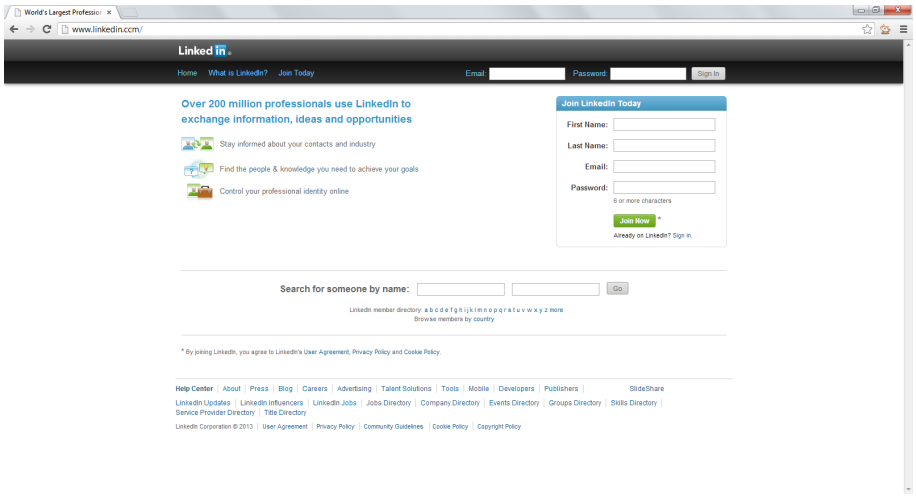

Figure C.6: Fake Linkedin site

C.4 Twitter

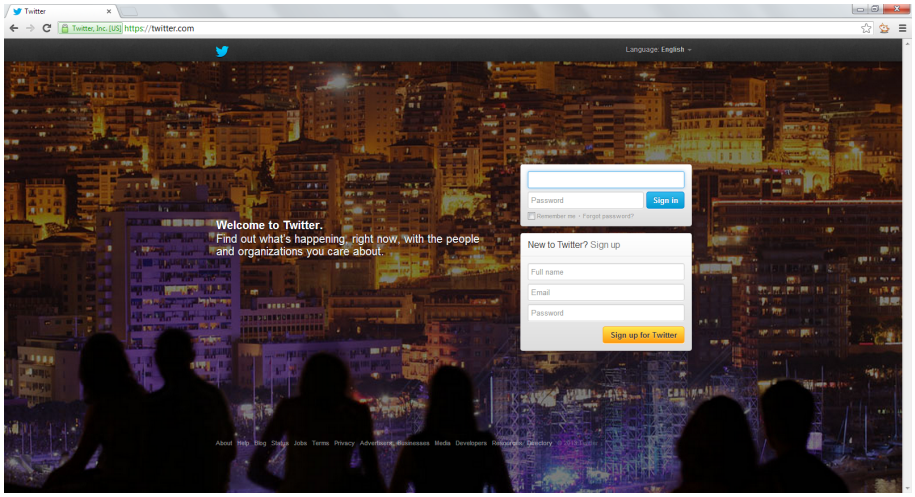

Figure C.7: Real Twitter site

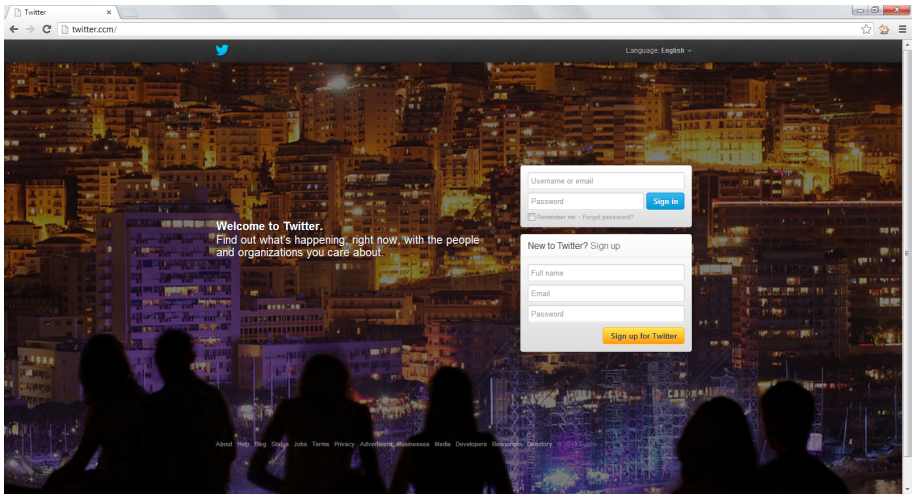

Figure C.8: Fake Twitter site

Appendix D

Interview Guide

The guide on the following paged is the one used in the interviews. The first page is an introduction to the interview with a place for the interviewee to give the informed consent. It is also read aloud by the interviewer. The second page is a description of the test, and the last is the questions for to use for the interviewer. it is translated into English in section 5.1.4. Some of the questions depend on how the interviewee do the user test. The guide is explained in section 5.1.5.

Petname test og undersøkelse

Tusen takk for at du har sagt deg villig til å delta i denne undersøkelsen. Mitt navn er Kent Varmedal og ønsker nå å la deg få teste ut en teknisk løsning for å forhindre at personlig informasjon kommer på avveie, samt evaluere den og snakke litt om phishing angrep.

Denne undersøkelsen skal ikke ta mer enn 20 minutter. Sesjonen vil bli tatt opp siden jeg ikke ønsker å gå glipp av noen av din kommentarer. Selv om jeg vil gjøre notater underveis, er jeg ikke på langt nær rask nok til å skrive alt. Så pass på å snakk høyt og tydelig.

Alle innspill vil bli behandlet konfidensielt. Det vil i praksis si at det er kun jeg som vil behandle informasjonen, og den informasjonen som blir brukt i master-oppgaven vil ikke kunne brukes til å identifisere noen av deltakerne i undersøkelsen.

Husk du trenger ikke å svare om du ikke ønsker og kan avslutte undersøkelsen når som helst.

Har du noen spørsmål angående det jeg nettopp har fortalt?

Er du villig til å delta i dette intervjuet?

_____ _____

Dato Navn

Test

Nå skal du få teste en teknisk løsning for å forhindre at falske nettsider stjeler personlig informasjon. Du skal logge inn på fire kjente nettsteder med et brukernavn og passord oppgitt til deg. Denne brukeren eksisterer ikke, men prosessen vil være tilstrekkelig for å få testet systemet. Du vil i tre runder få utlevert fire nettadresser. Disse adressene skal aksesseres i den oppgitte rekkefølgen, og er maskert for å hindre nysgjerrighet. Du vil bli observert mens du utfører disse.

Enheten koblet til datamaskinen er et eksternt Petname system. Det gir deg muligheten til å legge inn kallenavn for den siden du skal logge deg inn på. Et eksempel kan være «min bank» når man logger inn på «www.dnb.no»

I første runde skal du legge inn nye kallenavn for hver av sidene du får linker til. Trykk på linken og logg inn med det oppgitte brukernavnet og passordet. Legg inn nytt kallenavn for siden på enheten når du blir spurt om det.

Gjør det samme med de tre andre linkene.

I andre runde skal du logge inn på sidene på nytt med nye linker uten å bruke petname systemet.

I tredje og siste runde skal du logge inn på sidene, men nå med petname systemet tilkoblet.

Spørsmål til intervjuet

I runden uten OffPAD

- Virket alt normalt da du gikk til websidene?
- Hvis du merket noe unormalt, hva var det?

I runden med OffPAD

- Virket alt normalt da du gikk til websidene?
- Hvis du merket noe unormalt, hva var det?
- Ville du ha oppdaget phishing nettsiden uten OffPAD?
- Hvis du syntes alt var normalt, hvorfor la du ikke merke til advarselen fra OffPAD?

Generelle spørsmål rundt eksperimentet

- Hvordan ble din årvåkenhet endret etter å ha oppdaget en phishing side?
- Hvordan ble din følelse av sikkerhet endret når du benyttet OffPAD?
- Var du mer eller mindre oppmerksom under eksperimentet enn det du vanligvis er?
- Hvordan tror du bruk av OffPAD kan påvirke din daglige bruk av Internett?

Hvordan syns du det ville være å benytte OffPAD

Hvis den var integrert i smart-telefon?
Hvis den var en egen separat dings for petname autentisering?
Hvis den var en egen separat dings med flere sikkerhetsfunksjoner?

Normalt når du går til websider med innlogging, hvor oppmerksom er du på phishing angrep?

Hvordan vurderer du phishing som en trussel mot deg selv?

Hvordan synes du OffPAD prototypen var å bruke?

Er det noe annet du ønsker å legge til?

Jeg vil nå analysere de svarene du og andre har gitt meg, resultatet vil bli lagt frem i min master-oppgave.

Tusen takk for at du stilte opp.

Appendix E

Notification Form from Data Protection Official for Research

On the following pages you will find the filled out Notification Form from Data Protection Official for Research. It must be approved before the data gathering from the test is allowed to be done.

It is in Norwegain because it is required to be in Norwegian if the applicant is.

Norsk samfunnsvitenskapelig datatjeneste AS
NORWEGIAN SOCIAL SCIENCE DATA SERVICES

MELDESKJEMA

Meldeskjema (versjon 1.4) for forsknings- og studentprosjekt som medfører meldeplikt eller konsesjonsplikt (jf. personopplysningsloven og helseregisterloven med forskrifter).

1. Prosjekttittel		
Tittel	Brukertest og undersøkelse av phishing	
2. Behandlingsansvarlig institusjon		
Institusjon	Universitetet i Oslo	Velg den institusjonen du er tilknyttet. Alle nivå må oppgis. Ved studentprosjekt er det studentens tilknytning som er avgjørende. Dersom institusjonen ikke finnes på listen, vennligst ta kontakt med personvernombudet.
Avdeling/Fakultet	Det matematisk-naturvitenskapelige fakultet	
Institutt	Institutt for informatikk	
3. Daglig ansvarlig (forsker, veileder, stipendiat)		
Fornavn	Audun	Før opp navnet på den som har det daglige ansvaret for prosjektet.Veileder er vanligvis daglig ansvarlig ved studentprosjekt. Veileder og student må være tilknyttet samme institusjon. Dersom studenten har ekstern veileder, kan biveileder eller fagansvarlig ved studiestedet stå som daglig ansvarlig.Arbeidssted må være tilknyttet behandlingsansvarlig institusjon, f.eks. underavdeling, institutt etc. NB! Det er viktig at du oppgir en e-postadresse som brukes aktivt. Vennligst gi oss beskjed dersom den endres.
Etternavn	Jøssang	
Akademisk grad	Doktorgrad	
Stilling	Professor	
Arbeidssted	Institutt for Informatikk, Universitetet i Oslo	
Adresse (arb.sted)	Ole-Johan Dahls hus, Gaustadalléen 23b	
Postnr/sted (arb.sted)	0373 Oslo	
Telefon/mobil (arb.sted)	22845524 /	
E-post	josang@matnat.uio.no	
4. Student (master, bachelor)		
Studentprosjekt	Ja ● Nei ○	NB! Det er viktig at du oppgir en e-postadresse som brukes aktivt. Vennligst gi oss beskjed dersom den endres.
Fornavn	Kent Are	
Etternavn	Varmedal	
Akademisk grad	Lavere grad	
Privatadresse	Sinsenveien 5A	
Postnr/sted (privatadresse)	0572 Oslo	
Telefon/mobil	48002214 /	
E-post	kentav@ifi.uio.no	
5. Formålet med prosjektet		
Formål	Formålet er å teste og vurdere ekstern enhet med et "Petname System" for å forhindre phishing angrep på Internett.	Redegjør kort for prosjektets formål, problemstilling, forskningsspørsmål e.l. Maks 750 tegn.
6. Prosjektomfang		
Velg omfang	● Enkel institusjon ○ Nasjonalt samarbeidsprosjekt ○ Internasjonalt samarbeidsprosjekt	Med samarbeidsprosjekt menes prosjekt som gjennomføres av flere institusjoner samtidig, som har samme formål og hvor personopplysninger utveksles.
Oppgi øvrige institusjoner		
Oppgi hvordan samarbeidet foregår		
7. Utvalgsbeskrivelse		

Utvalget	Studenter med generell sikkerhetsinteresse.	Med utvalg menes dem som deltar i undersøkelsen eller dem det innhentes opplysninger om. F.eks. et representativt utvalg av befolkningen, skoleelever med lese- og skrivevansker, pasienter, innsatte.
Rekruttering og trekking	Frivillige studenter som blir oppfordret til å delta. Studentene kan få høre om prosjektet på slutten av en forelesing eller bli direkte spurt om de ønsker å delta.	Beskriv hvordan utvalget trekkes eller rekrutteres og oppgi hvem som foretar den. Et utvalg kan trekkes fra registre som f.eks. Folkeregisteret, SSB-registre, pasientregistre, eller det kan rekrutteres gjennom f.eks. en bedrift, skole, idrettsmiljø, eget nettverk.
Førstegangskontakt	Studenter kan melde sin egen interesse for å delta, eller studenter kan bli spurt av Kent Varmedal om de vil stille opp.	Beskriv hvordan førstegangskontakten opprettes og oppgi hvem som foretar den. Les mer om dette på temasidene Hva skal du forske på?
Alder på utvalget	□ Barn (0-15 år) □ Ungdom (16-17 år) ■ Voksne (over 18 år)	
Antall personer som inngår i utvalget	6	
Inkluderes det myndige personer med redusert eller manglende samtykkekompetanse?	Ja ○ Nei ●	Begrunn hvorfor det er nødvendig å inkludere myndige personer med redusert eller manglende samtykkekompetanse. Les mer om Pasienter, brukere og personer med redusert eller manglende samtykkekompetanse
Hvis ja, begrunn		
8. Metode for innsamling av personopplysninger		
Kryss av for hvilke datainnsamlingsmetoder og datakilder som vil benyttes	□ Spørreskjema ■ Personlig intervju □ Gruppeintervju ■ Observasjon □ Psykologiske/pedagogiske tester □ Medisinske undersøkelser/tester □ Journaldata □ Registerdata □ Annen innsamlingsmetode	Personopplysninger kan innhentes direkte fra den registrerte f.eks. gjennom spørreskjema, intervju, tester, og/eller ulike journaler (f.eks. elevmapper, NAV, PPT, sykehus) og/eller registre (f.eks. Statistisk sentralbyrå, sentrale helseregistre).
Annen innsamlingsmetode, oppgi hvilken		
Kommentar	Først vil deltakeren få teste ut systemet, for å så bli intervjuet med spørsmål til testen og phishing generelt.	
9. Datamaterialets innhold		
Redegjør for hvilke opplysninger som samles inn	Det vil bli observert hvordan den tekniske enheten blir benyttet, for å se etter brukermønster og hvor eventuelle problemer oppstår. Det er utarbeidet en intervjuguide for dette forsøket.	Spørreskjema, intervju-/temaguide, observasjonsbeskrivelse m.m. sendes inn sammen med meldeskjemaet. NB! Vedleggene lastes opp til sist i meldeskjema, se punkt 16 Vedlegg.
Samles det inn direkte personidentifiserende opplysninger?	Ja ○ Nei ●	Dersom det krysses av for ja her, se nærmere under punkt 11 Informasjonssikkerhet. Les mer om hva personopplysninger er NB! Selv om opplysningene er anonymiserte i oppgave/rapport, må det krysses av dersom direkte og/eller indirekte personidentifiserende opplysninger innhentes/registreres i forbindelse med prosjektet.
Hvis ja, hvilke?	□ 11-sifret fødselsnummer □ Navn, fødselsdato, adresse, e-postadresse og/eller telefonnummer	
Spesifiser hvilke		
Samles det inn indirekte personidentifiserende opplysninger?	Ja ○ Nei ●	En person vil være indirekte identifiserbar dersom det er mulig å identifisere vedkommende gjennom

Hvis ja, hvilke?		bakgrunnsopplysninger som for eksempel bostedskommune eller arbeidsplass/skole kombinert med opplysninger som alder, kjønn, yrke, diagnose, etc.
Samles det inn sensitive personopplysninger?	Ja ○ Nei ●	
Hvis ja, hvilke?	□ Rasemessig eller etnisk bakgrunn, eller politisk, filosofisk eller religiøs oppfatning □ At en person har vært mistenkt, siktet, tiltalt eller dømt for en straffbar handling □ Helseforhold □ Seksuelle forhold □ Medlemskap i fagforeninger	
Samles det inn opplysninger om tredjeperson?	Ja ○ Nei ●	Med opplysninger om tredjeperson menes opplysninger som kan spores tilbake til personer som ikke inngår i utvalget. Eksempler på tredjeperson er kollega, elev, klient, familiemedlem.
Hvis ja, hvem er tredjeperson og hvilke opplysninger registreres?		
Hvordan informeres tredjeperson om behandlingen?	□ Skriftlig □ Muntlig □ Informeres ikke	
Informeres ikke, begrunn		
10. Informasjon og samtykke		
Oppgi hvordan utvalget informeres	□ Skriftlig ■ Muntlig □ Informeres ikke	Vennligst send inn informasjonsskrivet eller mal for muntlig informasjon sammen med meldeskjema. NB! Vedlegg lastes opp til sist i meldeskjemaet, se punkt 16 Vedlegg. Dersom utvalget ikke skal informeres om behandlingen av personopplysninger må det begrunnes. Les mer om krav til samtykke
Begrunn		
Oppgi hvordan samtykke fra utvalget innhentes	■ Skriftlig ■ Muntlig □ Innhentes ikke	Dersom det innhentes skriftlig samtykke anbefales det at samtykkeerklæringen utformes som en svarslipp eller på eget ark. Dersom det ikke skal innhentes samtykke, må det begrunnes.
Innhentes ikke, begrunn		
11. Informasjonssikkerhet		
Direkte personidentifiserende opplysninger erstattes med et referansenummer som viser til en atskilt navneliste (koblingsnøkkel)	Ja ○ Nei ●	Har du krysset av for ja under punkt 9 Datamaterialets innhold må det merkes av for hvordan direkte personidentifiserende opplysninger registreres. NB! Som hovedregel bør ikke direkte personidentifiserende opplysninger registreres sammen med det øvrige datamaterialet.
Hvordan oppbevares navnelisten/ koblingsnøkkelen og hvem har tilgang til den?		
Direkte personidentifiserende opplysninger oppbevares sammen med det øvrige materialet	Ja ○ Nei ●	
Hvorfor oppbevares direkte personidentifiserende opplysninger sammen med det øvrige datamaterialet?		
Oppbevares direkte personidentifiserbare opplysninger på andre måter?	Ja ○ Nei ●	
Spesifiser		

Hvordan registreres og oppbevares datamaterialet?	□ Fysisk isolert datamaskin tilhørende virksomheten □ Datamaskin i nettverkssystem tilhørende virksomheten □ Datamaskin i nettverkssystem tilknyttet Internett tilhørende virksomheten □ Fysisk isolert privat datamaskin ■ Privat datamaskin tilknyttet Internett □ Videoopptak/fotografi ■ Lydopptak ■ Notater/papir □ Annen registreringsmetode	Merk av for hvilke hjelpemidler som benyttes for registrering og analyse av opplysninger. Sett flere kryss dersom opplysningene registreres på flere måter.
Annen registreringsmetode beskriv		
Behandles lyd-/videoopptak og/eller fotografi ved hjelp av datamaskinbasert utstyr?	Ja ● Nei ○	Kryss av for ja dersom opptak eller foto behandles som lyd-/bildefil. Les mer om behandling av lyd og bilde.
Hvordan er datamaterialet beskyttet mot at uvedkommende får innsyn?	Datamaterialet blir liggende på en bærbar PC med brukernavn og passord. Datamatrialet på PCen vil bli lagret i en kryptert pakket fil.	Er f.eks. datamaskintilgangen beskyttet med brukernavn og passord, står datamaskinen i et låsbart rom, og hvordan sikres bærbare enheter, utskrifter og opptak?
Dersom det benyttes mobile lagringsenheter (bærbar datamaskin, minnepenn, minnekort, cd, ekstern harddisk, mobiltelefon), oppgi hvilke	Selve opptaket vil bli tatt opp med en mobiltelefon for å så overføres til passordbeskyttet bærbar PC. Og vil bli slettet etter at opptakene er transkribert og anonymisert.	NB! Mobile lagringsenheter bør ha mulighet for kryptering.
Vil medarbeidere ha tilgang til datamaterialet på lik linje med daglig ansvarlig/student?	Ja ○ Nei ●	
Hvis ja, hvem?		
Overføres personopplysninger ved hjelp av e-post/Internett?	Ja ○ Nei ●	F.eks. ved bruk av elektronisk spørreskjema, overføring av data til samarbeidspartner/databehandler mm.
Hvis ja, hvilke?		
Vil personopplysninger bli utlevert til andre enn prosjektgruppen?	Ja ○ Nei ●	
Hvis ja, til hvem?		
Samles opplysningene inn/behandles av en databehandler?	Ja ○ Nei ●	Dersom det benyttes eksterne til helt eller delvis å behandle personopplysninger, f.eks. Questback, Synovate MMI, Norfakta eller transkriberingsassistent eller tolk, er dette å betrakte som en databehandler. Slike oppdrag må kontraktsreguleres Les mer om databehandleravtaler her
Hvis ja, hvilken?		
12. Vurdering/godkjenning fra andre instanser		
Søkes det om dispensasjon fra taushetsplikten for å få tilgang til data?	Ja ○ Nei ●	For å få tilgang til taushetsbelagte opplysninger fra f.eks. NAV, PPT, sykehus, må det søkes om dispensasjon fra taushetsplikten. Dispensasjon søkes vanligvis fra aktuelt departement. Dispensasjon fra taushetsplikten for helseopplysninger skal for alle typer forskning søkes Regional komité for medisinsk og helsefaglig forskningsetikk
Kommentar		
Søkes det godkjenning fra andre instanser?	Ja ○ Nei ●	F.eks. søke registereier om tilgang til data, en ledelse om tilgang til forskning i virksomhet, skole, etc.
Hvis ja, hvilke?		
13. Prosjektperiode		

Prosjektperiode	Prosjektstart:06.03.2013 Prosjektslutt:13.03.2013	Prosjektstart Vennligst oppgi tidspunktet for når førstegangskontakten med utvalget opprettes og/eller datainnsamlingen starter. Prosjektslutt Vennligst oppgi tidspunktet for når datamaterialet enten skal anonymiseres/slettes, eller arkiveres i påvente av oppfølgingsstudier eller annet. Prosjektet anses vanligvis som avsluttet når de oppgitte analyser er ferdigstilt og resultatene publisert, eller oppgave/avhandling er innlevert og sensurert.
Hva skal skje med datamaterialet ved prosjektslutt?	■ Datamaterialet anonymiseres □ Datamaterialet oppbevares med personidentifikasjon	Med anonymisering menes at datamaterialet bearbeides slik at det ikke lenger er mulig å føre opplysningene tilbake til enkeltpersoner.NB! Merk at dette omfatter både oppgave/publikasjon og rådata. Les mer om anonymisering
Hvordan skal datamaterialet anonymiseres?	Transkiberingen av intervjuene vil anonymiseres når de skrives, ved å bytte ut navnte navn og steder med enkeltbokstaver. Sittatene benyttet i masteroppgaven vil være korte og tatt ut av identifiserende kontekst.	Hovedregelen for videre oppbevaring av data med personidentifikasjon er samtykke fra den registrerte. Årsaker til oppbevaring kan være planlagte oppfølgningsstudier, undervisningsformål eller annet. Datamaterialet kan oppbevares ved egen institusjon, offentlig arkiv eller annet. Les om arkivering hos NSD
Hvorfor skal datamaterialet oppbevares med personidentifikasjon?		
Hvor skal datamaterialet oppbevares, og hvor lenge?		
14. Finansiering		
Hvordan finansieres prosjektet?	Dette gjøres av masterstudent.	
15. Tilleggsopplysninger		
Tilleggsopplysninger		
16. Vedlegg		
Antall vedlegg	1	

Appendix F

Reply from Data Protection Official for Research

The reply from Data Protection Official for Research approved the descried handling of the data from the user test.

The letter is on the following pages.

Norsk samfunnsvitenskapelig datatjeneste AS
NORWEGIAN SOCIAL SCIENCE DATA SERVICES

Harald Hårfagres gate 29
N-5007 Bergen
Norway
Tel: +47-55 58 21 17
Fax: +47-55 58 96 50
nsd@nsd.uib.no
www.nsd.uib.no
Org.nr. 985 321 884

Audun Jøsang
Institutt for informatikk
Universitetet i Oslo
Postboks 1080 Blindern
0316 OSLO

Vår dato: 07.03.2013 Vår ref:33673 / 3 / KH Deres dato: Deres ref:

TILBAKEMELDING PÅ MELDING OM BEHANDLING AV PERSONOPPLYSNINGER

Vi viser til melding om behandling av personopplysninger, mottatt 04.03.2013. Meldingen gjelder prosjektet:

33673	*Brukertest og undersøkelse av phishing*
Behandlingsansvarlig	*Universitetet i Oslo, ved institusjonens øverste leder*
Daglig ansvarlig	*Audun Jøsang*
Student	*Kent Are Varmedal*

Personvernombudet har vurdert prosjektet og finner at behandlingen av personopplysninger er meldepliktig i henhold til personopplysningsloven § 31. Behandlingen tilfredsstiller kravene i personopplysningsloven.

Personvernombudets vurdering forutsetter at prosjektet gjennomføres i tråd med opplysningene gitt i meldeskjemaet, korrespondanse med ombudet, eventuelle kommentarer samt personopplysningsloven og helseregisterloven med forskrifter. Behandlingen av personopplysninger kan settes i gang.

Det gjøres oppmerksom på at det skal gis ny melding dersom behandlingen endres i forhold til de opplysninger som ligger til grunn for personvernombudets vurdering. Endringsmeldinger gis via et eget skjema http://www.nsd.uib.no/personvern/meldeplikt/skjema.html. Det skal også gis melding etter tre år dersom prosjektet fortsatt pågår. Meldinger skal skje skriftlig til ombudet.

Personvernombudet har lagt ut opplysninger om prosjektet i en offentlig database, http://pvo.nsd.no/prosjekt.

Personvernombudet vil ved prosjektets avslutning, 01.04.2013, rette en henvendelse angående status for behandlingen av personopplysninger.

Vennlig hilsen

Vigdis Namtvedt Kvalheim

Kjersti Håvardstun

Kjersti Håvardstun tlf: 55 58 29 53
Vedlegg: Prosjektvurdering
Kopi: Kent Are Varmedal, Sinsenveien 5A, 0572 OSLO

Personvernombudet for forskning

Prosjektvurdering - Kommentar

Prosjektnr: 33673

Personvernombudet finner informasjonsskrivet til utvalget tilfredsstillende utformet forutsatt at følgende tilføyes:
- dato for forventet prosjektslutt
- navn og kontaktopplysninger til daglig ansvarlig for studien

Prosjektet skal avsluttes 13.03.2013 og innsamlede opplysninger skal da anonymiseres og lydopptak slettes. Anonymisering innebærer at direkte personidentifiserende opplysninger som navn/koblingsnøkkel slettes, og at indirekte personidentifiserende opplysninger (sammenstilling av bakgrunnsopplysninger som f.eks. yrke, alder, kjønn) fjernes eller grovkategoriseres slik at ingen enkeltpersoner kan gjenkjennes i materialet.